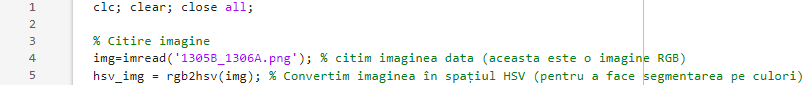
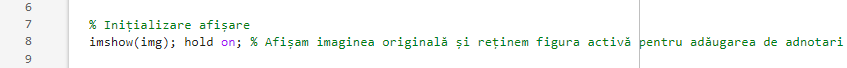
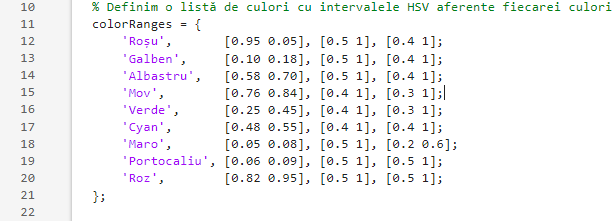
***Tema 2 SVA***

**Pasul 1:** Citim imaginea data si o convertim din RGB in HSV deoarece o imagine hsv este mai utilă pentru detectarea culorilor.





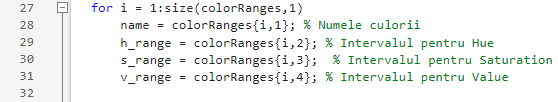
**Pasul 2:** Definim matricea “colorRanges” deoarece pentru fiecare culoare vom avea un interval de valori pentru hue, saturation și value. Aceasta matrice o vom folosi la segmentarea obiectelor colorate din imagine. Valorile din matrice nu sunt standard, ci au fost luate din mai multe combinatii din surse online. (Ex: Pentru roșu avem [0.95 0.05] deoarece roșul e la capătul și începutul cercului hue, iar in HSV, hue = 0 și hue = 1 înseamnă aceeași culoare adică roșu).



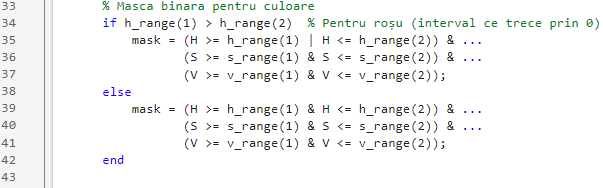
**Pasul 3:** Avand mai sus imaginea HSV (aceasta fiind o matrice 3D), vom extrage din ea fiecare canal (hue, saturation & value - fiecare fiind o matrice 2D) deoarece le vom folosi separat pentru a face detecția culorilor în funcție de pragurile HSV.



**Pasul 4:** Ne vom folosi de matricea “colorRanges” parcurgand-o cu un for de la 1 pâna la nr de culori din matrice cu scopul de a extrage pragurile de culoare pentru fiecare canal (hue, saturation & value) din matricea initiala.



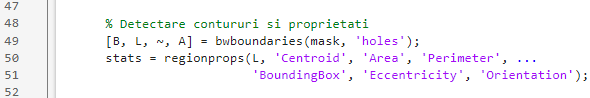
**Pasul 5:** Dorim să construim o mască binară, adică o matrice 2D de dimensiunea imaginii, unde pixelii care respectă condițiile de culoare sunt true (1), iar ceilalți sunt false (0). Însă avem o problemă specială la intervalul Hue și anume canalul hue este circular: 0 și 1 reprezintă aceeași nuanță. Deci dacă prima valoare h\_range(1) > a doua valoare h\_range(2), avem o zonă care traversează începutul scării hue, cum e la roșu și trebuie să unim două intervale. Altfel avem un interval „normal” care nu trece peste 0.



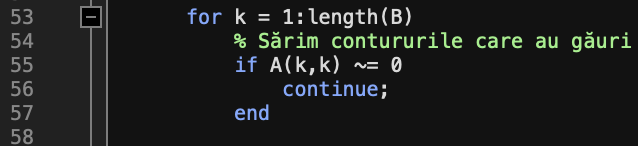
**Pasul 6:** Ulterior eliminăm obiectele mici din mască (eg. zgomot) și păstrăm doar obiectele cu peste 300 pixeli. Pastram totodată masca originală pentru a verifica dacă obiectul contine găuri și completam găurile obiectelor, pentru detectarea corectă a formei.



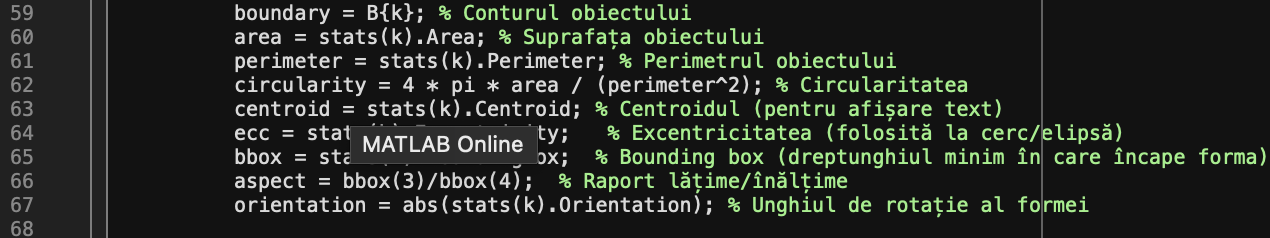
**Pasul 7:** Pentru a detecta contururile folosim funcția “bwboundaries” aplicată pe imaginea binara “mask” și totodată detectăm și găurile din obiect. **B** v-a contine contururile detectate, **L** este matricea de etichete în care pixelii aparținând aceleiași regiuni au aceeași valoare, iar **A** este matricea de adiacență care indică relațiile dintre forme și găurile din ele. Apoi extragem proprietățile geometrice pentru fiecare obiect detectat: centru, arie, perimetru, încadrare, excentricitate, orientare folosind functia “regionprops”



**Pasul 8:** Parcurgem contururile fiecărui obiect și sarim mai departe daca gasim unul cu o gaura interna.



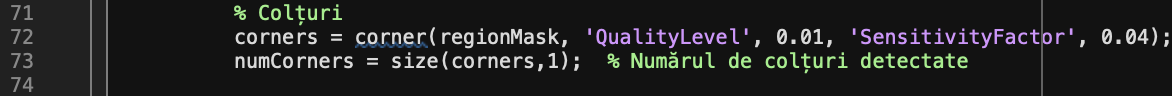
**Pasul 9:** Extragem fiecare caracteristică și particularitate a obiectului.



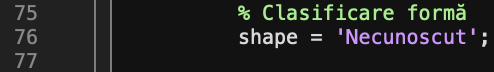
**Pasul 10:** Extragem o mască binară care izolează doar regiunea curentă (k) din imagine deoarece noi vrem sa procesam doar o anumita zona (k), nu toata imaginea.

****

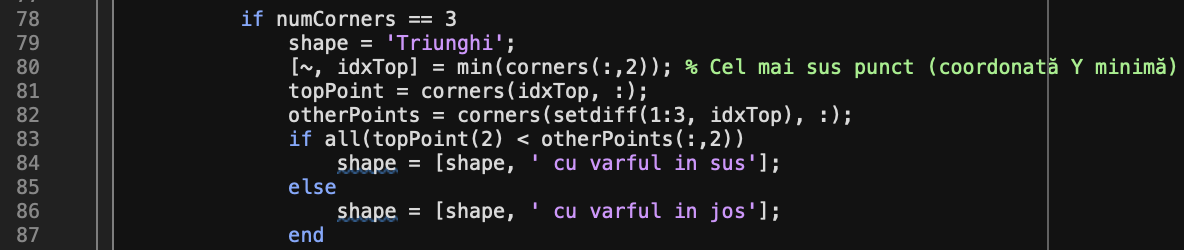
**Pasul 11:** Vom folosi funcția “corner” aplicată pe masca creata mai suspentru a detecta colțurile formei geometrice și le vom și număra.



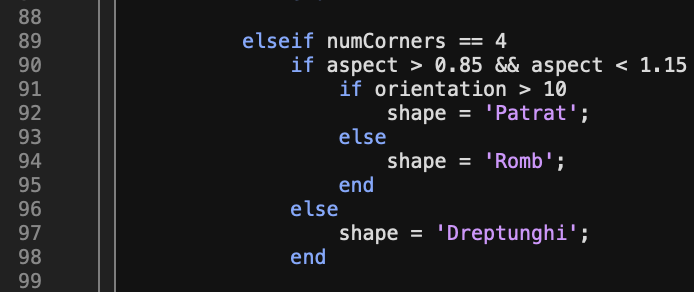
**Pasul 12:** Acum începem procesul de clasificare a fiecărei forme geometrice. Inițial initializam forma ca fiind “necunoscută”.



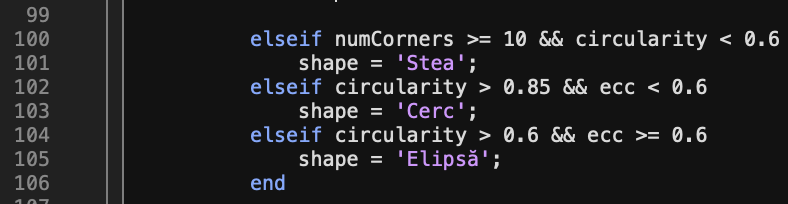
**Pasul 13:** Prima data verificam daca obiectul are 3 colțuri, asta înseamnă ca sigur este triunghi și îl etichetam. Apoi, extragem coordonatele colțului superior și verificăm dacă punctul extras este mai sus decât celelalte colțuri rămase, asta înseamnă ca este triunghi cu varful in sus. Altfel este triunghi cu vârful în jos.



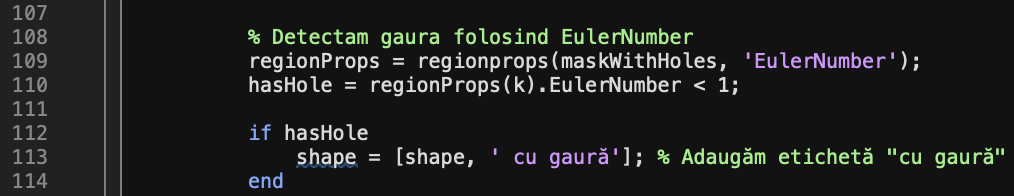
**Pasul 14:** În al doilea rand verificam daca obiectul are 4 colțuri, înseamnă ca este pătrat, romb sau dreptunghi. Dacă raportul dintre inaltime si latime (aspect) este aproape 1 înseamnă ca avem un patrat sau romb. Dacă acesta este înclinat mai mult de 10 grade atunci este romb, altfel este patrat. Dacă nu întrunește caracteristicile de mai sus, atunci este dreptunghi.



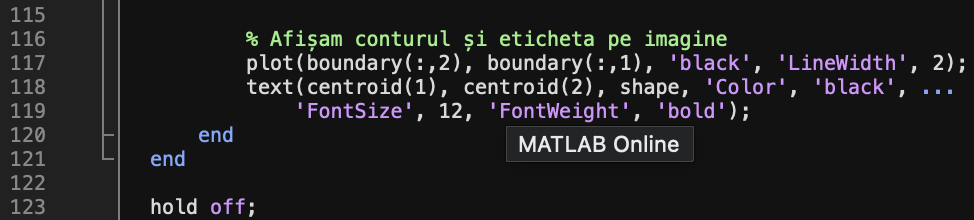
**Pasul 15:** Nu in ultimul rand, mai avem de clasificat și celelalte obiecte rămase din imagine. Dacă sunt multe colțuri, dar forma nu e rotundă, atunci obiectul este o stea. Daca forma e rotunda dar nu e alungita atunci e cerc, altfel este elipsa.



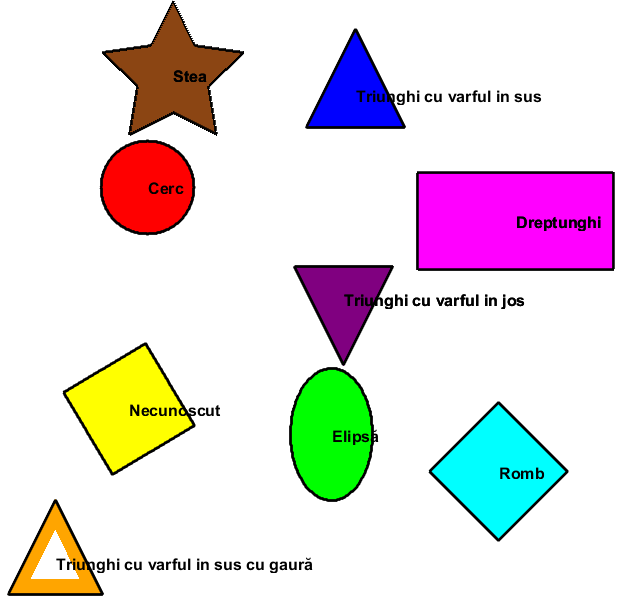
**Pasul 16:** Folosind masca originala de la început cu gauri, ne folosim de “Numărul lui Euler” pentru a vedea cate găuri are obiectul. Dacă “Numărul lui Euler” este mai mic decat 1 înseamnă ca obiectul are cel putin o gaura si atunci etichetam obiectul intocmai.



**Pasul 17:** In final, avand imaginea originală cu obiectele, trasam conturul formelor geometrice cu negru pentru a fi mai bine evidentiate și le etichetam pe fiecare cu forma detectata de algoritmul nostru.



**Imaginea finala:**

****