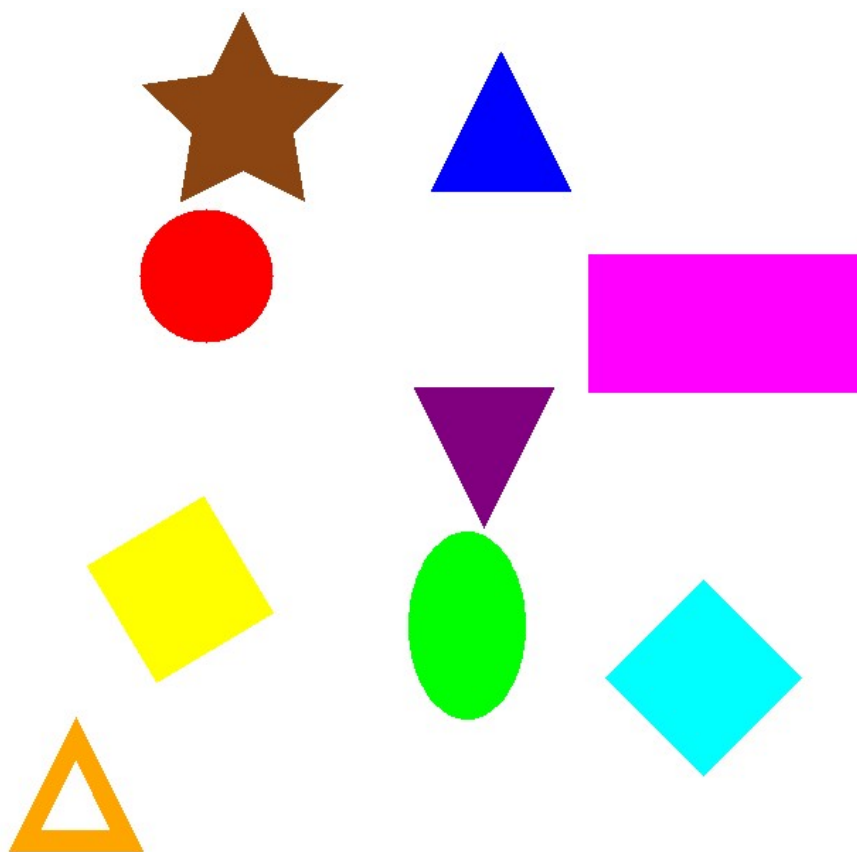


Tema 2 SVA

Mihai Tudor

1305B

Imagine tema



Script

```
img = imread("imagine_tema.png");

imshow(img); title("Imagine originala - 9 forme");

% Preprocessing

gray = rgb2gray(img);

bw = imbinarize(gray);

bw = imcomplement(bw);    %formele devin albe, fundalul negru

bw = imfill(bw, 'holes');    %umplere gauri din forme

[B,L] = bwboundaries(bw, 'noholes');

stats = regionprops(L, 'Centroid', 'Area', 'Perimeter', 'BoundingBox', 'Eccentricity', 'Solidity', 'Orientation');

figure; imshow(img); hold on; title("Imaginea cu forme detectate");

for k = 1:length(B)    %iteratie pentru fiecare forma detectata

    s = stats(k);

    boundary = B{k};

    bb = s.BoundingBox;

    circ = 4*pi*s.Area/(s.Perimeter^2);

    ar = bb(3)/bb(4);

    ecc = s.Eccentricity;

    % Detectie colturi: identifica numarul de colturi relevante

    [bx,by] = deal(max(1, ceil(bb(1))), max(1, ceil(bb(2))));

    [bw_,bh_] = deal(min(size(bw,2)-bx, floor(bb(3))), min(size(bw,1)-by, floor(bb(4))));

    corners = [];

    if bw_ > 0 && bh_ > 0

        region = bw(by:by+bh_, bx:bx+bw_);

        hc = detectHarrisFeatures(region);

        if ~isempty(hc) && size(hc.Location,1) >= 3
```

```

pts = hc.selectStrongest(10).Location + [bx-1, by-1];

corners = clusterCorners(pts,15);

else

N = size(boundary,1); pts = [];

for i = 11:5:N-10

    pa = boundary(i-10,:); pc = boundary(i,:); pb = boundary(i+10,:);

    a = acosd(min(1,max(-1,dot(pa-pc,pb-pc)/(norm(pa-pc)*norm(pb-pc)))));

    if a < 140, pts = [pts; pc]; end

end

corners = clusterCorners(pts,15);

end

nc = size(corners,1);

%extragere culori: identifica culoarea dominanta a formei

mask = L == k; nz = sum(mask(:));

rgb = squeeze(sum(reshape(double(img) .* mask, [], 3), 1)) / nz;

[~, idx] = max(rgb);

colors = {'red', 'green', 'blue'};

dom = colors{idx};

[r,g,b] = deal(rgb(1), rgb(2), rgb(3));

%clasificare forme

label = 'forma necunoscuta';

if nc >= 5 && s.Solidity < 0.8 && r > 80 && g < 100 && b < 100, label = 'stea';

elseif circ > 0.85 && ecc < 0.6 && r > 150 && g < 100 && b < 100, label = 'cerc'; %circularitate mare,
excentricitate mica

elseif circ > 0.8 && ecc > 0.6 && g > 150 && r < 100 && b < 100, label = 'elipsa'; %circularitate mare,
excentricitate mare

elseif nc >= 3 && nc <= 4 && circ < 0.7, label = 'triunghi'; end

% detectie dupa culoarea formelor in imagine - caz specific

```

```

if b > 150 && r < 100 && g < 100, label = 'triunghi'; end
if r > 50 && r < 120 && b > 100 && g < 70, label = 'triunghi'; end
if r > 200 && b > 100 && g < 100, label = 'dreptunghi'; end
if r > 200 && g > 200 && b < 50, label = 'patrat'; end
if b > 150 && g > 150 && r < 50, label = 'romb'; end

%desenare si adnotare
rectangle('Position', bb, 'EdgeColor', 'k', 'LineWidth', 1);
text(s.Centroid(1), s.Centroid(2), label, 'Color', 'k', 'FontSize', 11, 'FontWeight', 'bold');
end

%functie cluster care grupeaza punctele pentru a evita numararea aceluiasi colt de mai multe ori
function clustered = clusterCorners(corners, t)
if isempty(corners), clustered = []; return; end
clustered = []; used = false(size(corners,1),1);
for i = 1:size(corners,1)
    if used(i), continue; end
    used(i) = true; cluster = corners(i,:);
    for j = find(~used)
        if norm(corners(i,:) - corners(j,:)) < t
            cluster = [cluster; corners(j,:)]; used(j) = true;
        end
    end
end
clustered = [clustered; mean(cluster,1)];
end
end

```

Imaginea cu forme detectate

