

```

1 function [] = testImages()
2
3     aux = 1;
4     accuracy_overall_stack = [];
5     targetToText = ["circle" "kite" "parallelogram"
6                     "square" "trapezoid" "triangle"];
7     imgres = [25 25];
8     inputs = [];
9     k = 0;
10    target = zeros(6, 60);
11
12
13    for i = 1:length(targetToText)
14        Directory = './NN_Tema1_images\test\' + targetToText(i);
15        % Read images from Images folder
16        images = dir(fullfile(Directory, '*.png'));
17
18        for k = (k+1):length(images)*i
19            target(i, k) = 1;
20        end
21
22        for j = 1:length(images)
23            image = imread(fullfile(Directory, images(j).name)); % Read image
24            image = imresize(image, imgres);
25            binarizedImg = imbinarize(image);
26
27            imageVector1 = (binarizedImg( : ));
28
29            inputs( : , aux) = imageVector1;
30            aux = aux + 1;
31        end
32
33    end
34
35
36    load('Treinos\Train10.mat'); %carrega a rede neuronal
37    y = sim(net, inputs); % Simula
38
39    r=0;
40    for i=1:size(y,2) % Para cada classificacao
41        [a b] = max(y(:,i)); %b guarda a linha onde encontrou valor mais alto
42        [c d] = max(target(:,i)); %d guarda a linha onde encontrou valor mais alto
43        if b == d % se estao na mesma linha, a classificacao foi c
44            r = r+1;
45        end
46    end
47
48    accuracy = r/size(y,2)*100;
49    fprintf('Precisao total %f\n', accuracy);
50    accuracy_overall_stack = [accuracy_overall_stack accuracy];
51
52 end

```