

```

1 function trainimages()
2
3     aux = 1;
4     targetToText = ["circle" "kite"
5                     "parallelogram" "square"
6                     "trapezoid" "triangle"]; %Vetor com o nome das pastas de cada figura geométrica
7     imgres = [25 25]; %Tamanho para redimensionar a imagem
8     inputs = []; %Dados das imagens
9
10    nrEpochs = 10; %Número de repetições de cada combinação
11    accuracy_overall_stack = []; %Precisão total
12    accuracy_teste_stack = []; %Precisão de teste
13
14    target = zeros(6, 30); %Inicialização da matriz dos targets a zeros(varia consoar
15    %'start' --> target = zeros(6, 30); 6 = numeros de formas 30 = numeros de formas
16    %'test' --> target = zeros(6, 60);
17    %'train' --> target = zeros(6, 300);
18    %'all' --> target = zeros(6, 390); (todas as imagens das pastas 'start','test'
19
20    k = 0;
21
22    for i = 1:length(targetToText)
23        Directory = './NN_Tema1_images\start\' + targetToText(i); %Directoria de cada
24        images = dir(fullfile(Directory, '*.png')); %Lê imagens(.png) da pasta
25
26        for k = (k+1):length(images)*i %Ciclo para a construção dos targets
27            target(i, k) = 1;
28        end
29
30        for j = 1:length(images)
31            image = imread(fullfile(Directory, images(j).name)); %Lê imagem
32            image = imresize(image, imgres); %Redimensiona imagem
33            binarizedImg = imbinarize(image); %Binariza imagem (0/1)
34
35            imageVector1 = (binarizedImg( : ));
36
37            inputs( : , aux) = imageVector1; %Adiciona ao vetor dos inputs(Informação
38            aux = aux + 1;
39        end
40    end
41
42    for epoch = 1 : nrEpochs
43        net = feedforwardnet([10 10]); %Criação da rede neuronal
44        net.layers{1:end-1}.transferFcn = 'tansig'; %função ativação
45        net.layers{end}.transferFcn = 'tansig'; %função de saída
46        net.trainFcn = 'traingd'; %função de treino
47        net.divideFcn = 'dividerand'; %função de divisão
48        %parametros da função de divisão
49        net.divideParam.trainRatio = 0.7;
50        net.divideParam.valRatio = 0.15;
51        net.divideParam.testRatio = 0.15;
52
53        [net,tr] = train(net, inputs, target); %treina a rede
54        view(net); %Visualizar a rede
55        out = sim(net, inputs); %simula a rede
56
57        plotconfusion(target, out) % Matriz de confusao
58
59        % plotperf(tr) % Grafico com o desempenho da rede nos 3 conjuntos
60

```

```

61     %Calcula e mostra a percentagem de classificacoes corretas no total dos exem
62     r=0;
63     for i=1:size(out,2)           % Para cada classificacao
64         [a b] = max(out(:,i));    %b guarda a linha onde encontrou valor mais alt
65         [c d] = max(target(:,i)); %d guarda a linha onde encontrou valor mais alt
66         if b == d                 % se estao na mesma linha, a classificacao foi
67             r = r+1;
68         end
69     end
70
71     accuracy = r/size(out,2)*100;
72     fprintf('epoch: %d Precisao total %f\n', epoch, accuracy);
73     accuracy_overall_stack = [accuracy_overall_stack accuracy];
74
75     % simula a rede no conjunto teste
76     TInput = inputs(:, tr.testInd);
77     TTargets = target(:, tr.testInd);
78
79     out = sim(net, TInput);
80
81     %Calcula e mostra a percentagem de classificacoes corretas no conjunto de tes
82     r=0;
83     for i=1:size(tr.testInd,2)    % Para cada classificacao
84         [a b] = max(out(:,i));    % b guarda a linha onde encontrou valor mais
85         [c d] = max(TTargets(:,i)); % d guarda a linha onde encontrou valor mais
86         if b == d                 % se estao na mesma linha, a classificacao fo
87             r = r+1;
88         end
89
90     end
91     accuracy = r/size(tr.testInd,2)*100;
92     fprintf('epoch: %d Precisao teste %f\n\n', epoch, accuracy);
93
94     accuracy_teste_stack = [accuracy_teste_stack accuracy];
95
96 end
97
98 round(mean(accuracy_overall_stack));
99 round(mean(accuracy_teste_stack));
100
101 save('start.mat', 'net'); %guarda a rede treinada com o nome escolhido
102 end

```