



Trabalho de Implementação

O reconhecimento de padrões é o estudo de como as máquinas observam seu entorno, aprendem a distinguir padrões de interesse e tomam decisões razoáveis sobre as categorias dos padrões. Um padrão é uma descrição de um objeto. Um computador consegue reconhecer padrões, convertendo-os em sinais digitais e comparando-os com outros sinais já armazenados na memória.

Um sistema de reconhecimento geralmente compreende três componentes principais: pré-processamento, extração de características e classificação. Na etapa de pré-processamento, os dados de entrada são manipulados por uma variedade de métodos que fazem operações, tais como remoção de ruído, segmentação e melhoramento da qualidade dos mesmos. Na extração de características, o objetivo é representar os dados de entrada em termos de medidas quantificáveis que possam ser utilizados facilmente na etapa de classificação. E

O problema do reconhecimento de padrões é reconhecer padrões que sejam, em algum sentido, “os mesmos” apesar de ter experimentado uma variedade de transformações permitidas. Os padrões na vida real apresentam transformações geométricas lineares (rotação, escala e translação), deformações não lineares e variância de iluminação e background, etc. Este tipo de reconhecimento pode ser uma tarefa simples para os seres humanos e para os animais, mas converte-se em um grande problema se tentamos realizá-lo através de um computador. Os métodos tradicionais de reconhecimento de padrões carecem da habilidade para reconhecer o mesmo padrão com certo tipo de variância.

1. Dadas as imagens de números, gere 20 imagens com diferentes rotações para cada um dos números, use o comando `imrotate`.
2. Cada imagem rotacionada deve ser escalada em 0.5, 0.75, 1, 1.25, 1.5 e 2
3. Em cada imagem escalada deve ser inserido ruído tipo sal e pimenta usando a função `imnoise`, usar os seguintes valores : 0, 0.01, 0.02, 0.03
4. Depois extrair os momentos de Hu para cada imagem (código disponível no site)
5. Dividir a base em dois conjuntos, uma para teste e o outro para treino. Utilize a função `crossvalind()` para executar essa tarefa. Exemplo:

```
[train , test] = crossvalind('holdOut',etiqueta);
```

Divide a base em dois conjuntos, cada um corresponde ao 50% da base

6. Utilize a base de treino para “aprender” os diferentes padrões. Use o comando `classify()`

```
predict = classify(test_data , train_data , train_label);
```

onde `test_data` é o conjunto de teste, `train_data` o conjunto de treino e `train_label` as etiquetas do conjunto de treinamento. A função retorna como resultado as etiquetas do conjunto de teste

O seguinte código permite realizar a leitura das imagens, com uma determinada extensão (“png”, “jpg”, etc), que se encontram dentro de um diretório.

```
endereco = 'D:\pdi\images';
ext = 'png';
% procura as imagens com extensão png dentro do diretório images
arquivos = dir([endereco '*' ext]); %ler as imagens
quant_img = length(arquivos);
for i = 1:quant_img
    img = imread([endereco , arquivos(i,1).name]);
    ...
end
```