

Trabalho Prático: A Implementação de um OCR simplificado

Data de entrega: 03/11/2021 até às 08:30 horas no Canvas

Valor: 45 pontos

Penalidade por atraso: 10 pontos ao dia

Grupos: 3 ou 4 componentes.

Descrição:

Neste trabalho, você deverá implementar um OCR simples, capaz de ler uma imagem contendo uma sequência de dígitos manuscritos e identificar o seu valor. As seguintes especificações mínimas devem ser consideradas:

- a) A imagem que contém a sequência terá formato JPG ou PNG, será monocromática com qualquer resolução e quantização.
- b) A imagem conterá apenas uma sequência, com qualquer quantidade de dígitos.
- c) A imagem poderá conter outros elementos gráficos.
- d) A cor dos dígitos é mais escura que o fundo. Inverta a paleta de tons, caso o fundo seja mais escuro.
- e) A sequência estará escrita em qualquer posição da imagem, em uma direção única e aproximadamente linear, fazendo um ângulo entre -30 e +30 graus em relação ao eixo x da imagem.



Exemplos da base MNIST

Especificações do programa:

- a) O programa deve ser implementado em C++, Python ou Java.
- b) É permitido o uso de funções elementares de bibliotecas como o OpenCV. Por função elementar entende-se uma função básica de manipulação de imagens, cujo resultado não seja a solução final do problema (valor do número). Ex: leitura de arquivos, cálculo de histogramas, filtros, cálculo de distâncias, projeções, características, classificadores.

- c) O OCR deverá oferecer as seguintes opções:
- Utilizando a base de dados MNIST¹:
 - i. Reamostrar o número de tons de cinza das imagens como achar mais conveniente.
 - ii. Aplicar transformações de rotação e escala se achar conveniente para o alinhamento dos dígitos
 - iii. Extrair as projeções verticais e horizontais dos dígitos das imagens para os grupos de treino e teste.
 - iv. Interpolarm as projeções para um número fixo de valores, entre 10 e 32 à escolha. O padrão será formado pela concatenação das projeções verticais e horizontais.
 - v. Treinar um classificador de Mahalanobis com os padrões da base de treino e testar com a base de teste
 - vi. Treinar uma SVM com os mesmos padrões
 - vii. Treinar uma rede neural não convolucional com os mesmos padrões
 - viii. Calcular a matriz de confusão de cada classificador e a acurácia obtida nos testes
 - Ler e visualizar uma nova imagem nos formatos JPG e PNG.
 - Exibir a imagem em uma janela.
 - Identificar a região da imagem que contém a sequência.
 - Separar os diversos dígitos da sequência e calcular suas projeções
 - Interpolarm as projeções para um número fixo de valores, entre 16 e 32 à escolha. O padrão ser formado pela concatenação das projeções verticais e horizontais.
 - Reamostrar o número de tons de cinza e/ou aplicar transformações como achar mais conveniente para aproximar as imagens com a base de dados MNIST
 - Classificar os dígitos segmentados usando os 3 classificadores
 - Informar o valor do número representado na sequência, conforme a saída de cada classificador.
- d) O tempo de execução de cada classificador deve ser medido e exibido na interface.

A documentação, em formato ABNT, deve conter:

- a) A descrição do problema e das técnicas implementadas para a solução.
- b) A ordem de complexidade dos métodos implementados.
- c) As referências das bibliotecas utilizadas e instruções para instalação, compilação e uso.
- d) As medidas de tempo de execução.
- e) Exemplos de erros e acertos dos métodos.
- f) Uma discussão comparativa dos resultados e das escolhas dos parâmetros dos classificadores.
- g) Referências bibliográficas.

O que entregar:

Arquivos fontes e documentação no padrão ABNT (não inclua a base MNIST). Coloque todos os arquivos na raiz de um diretório cujo nome deve ser o número de matrícula de um dos componentes. Comprima o diretório e poste no Canvas até a hora especificada . O tamanho total dos arquivos não deve ultrapassar 5 Mbytes.

¹ <http://yann.lecun.com/exdb/mnist/>

Observações:

1. O trabalho deverá ser feito em grupos de três ou quatro alunos, sem qualquer participação de outros grupos e/ou ajuda de terceiros nem utilização de trabalhos feitos em semestres anteriores ou códigos disponibilizados em qualquer meio. Grupos com mais que 4 alunos receberão apenas 50% da nota auferida.
2. Os arquivos de código postados devem conter um cabeçalho com a identificação da disciplina e dos componentes do grupo. Nomes não informados no cabeçalho não receberão nota. O código deve ser extensamente comentado, com a descrição da lógica usada. Arquivos pouco comentados e difíceis de ler terão sua nota reduzida em 30%.
3. Trabalhos iguais, na sua totalidade ou em partes, copiados, reaproveitados de semestres anteriores, por todos ou alguns dos componentes, “encomendados” ou outras barbaridades do gênero, receberão nota 0. É responsabilidade do aluno manter o sigilo sobre seu trabalho, evitando que outros alunos tenham acesso a ele, como por exemplo em sites de compartilhamento público de código. No caso de cópia, ambos os grupos serão penalizados, independentemente de quem lesou ou foi lesado no processo.
4. Durante a apresentação poderão ser feitas perguntas relativas ao trabalho, as quais serão consideradas para fim de avaliação. Todos os componentes devem comparecer e serem capazes de responder a quaisquer perguntas e/ou alterar o código de qualquer parte do trabalho. A avaliação será individual.
5. A avaliação será baseada nos seguintes critérios:
 - Correção e robustez dos programas
 - Conformidade às especificações
 - Clareza de codificação (comentários, endentação, escolha de nomes para identificadores)
 - Parametrização
 - Apresentação individual
 - Documentação
6. Trabalhos de qualidade superior poderão ganhar pontos extras!