

Trabalho de implementação 2

Instruções: este trabalho deve ser realizado preferencialmente em duplas. Os alunos têm liberdade para escolha da linguagem na qual irá implementar os exercícios propostos, mas é dada preferência à linguagem Python. Até o dia **29/04/2022**, cada aluno deverá fazer o *upload* no Moodle da disciplina de um arquivo .zip, contendo

- Os códigos fonte
- Um breve relatório (2-3 páginas) explicando as implementações e testes. Você pode adicionar imagens em um anexo (extrapolando o número de páginas indicado).

Alternativamente, você pode enviar um Jupyter notebook em Python contendo tanto o código quanto as explicações sobre os testes.

Enunciado: o problema de *Content-based image retrieval* (CBIR) consiste em fornecer uma imagem de consulta e recuperar imagens “semelhantes” em uma base de dados. O conceito de “semelhante” depende da aplicação, normalmente envolvendo informação de cor, textura e forma.

Neste trabalho, o objetivo é criar um sistema de CBIR usando informação de forma como medida de similaridade. Cada dupla deve desenvolver um sistema simples que receba como entrada uma imagem de busca, um número inteiro positivo N , e então retornar as N imagens da base de dados mais semelhantes à imagem de busca, em ordem decrescente de similaridade. Se a imagem de busca estiver na base de dados, idealmente ela deve ser a primeira imagem a ser recuperada.

Nos seus testes, considere a base de imagens binárias “The 216-Shape Database” fornecida em <http://vision.lcms.brown.edu/sites/default/files/216db.tar.gz>, que contém 216 imagens pertencentes a diferentes classes (tipicamente informadas na raiz do nome de cada imagem). Com base nas imagens fornecidas, pense em um conjunto de descritores de forma e uma métrica de distância ou similaridade a ser usado para comparar duas imagens. Em particular, pense em uma métrica que apresente alguma invariância com relação à variação de escala e rotações.

O relatório deve conter a descrição dos atributos e da métrica escolhidos, com **justificativas** (você não está limitado a usar os atributos e métricas estudados em aula, seja criativo). Como exemplo de execução, use uma das imagens `fork` do dataset como a consulta, e avalie quantas “versões” dela são recuperadas dentre as N mais similares. Além desse teste, coloque outros exemplos de buscas como ilustração – um exemplo hipotético de execução (para $N = 6$) é ilustrado na Figura 1, onde a imagem `bird02` é usada na consulta, e todas as imagens recuperadas são da classe `bird` – com `bird02` sendo a primeira recuperada.

CMP 197 apenas: escreva uma rotina que aplique uma variação aleatória de escala (com um fator selecionado no intervalo $[0.85, 1.15]$) e uma rotação aleatória tanto na imagem de consulta quanto nas imagens a serem buscadas. Para uma mesma imagem de consulta, avalie o quanto as imagens recuperadas foram afetadas por essas transformações. **Dica:** você pode usar as rotinas `getRotationMatrix2D` e `warpAffine` do OpenCV.

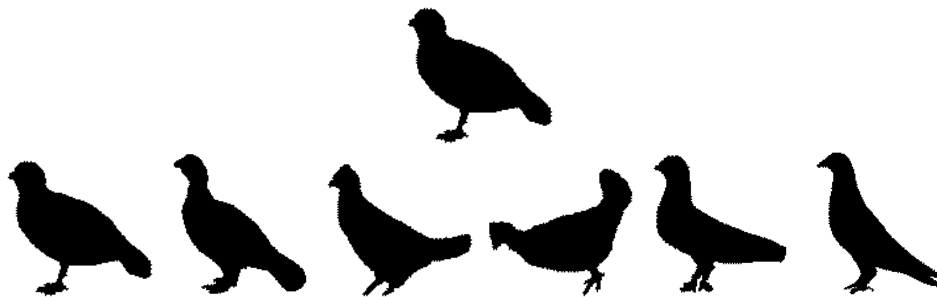


Figura 1: Possível exemplo de execução do sistema. Primeira linha: imagem de consulta. Segunda linha, da esquerda para a direita: $N = 6$ imagens recuperadas da base de dados, em ordem decrescente de similaridade.