

**Instituto Superior de Engenharia de Lisboa**  
**Codificação de Sinais Multimédia**

**2º Semestre de 2020/2021**

O ficheiro **Jupyter Notebook** com o relatório e código implementado deve submetido no Moodle até **6 de Junho**. O nome do ficheiro deve ter o seguinte formato: T41Gxx\_TP3.ipynb ou T42Gxx\_TP3.ipynb ou T4NGxx\_TP3.ipynb, consoante a turma que estiverem a frequentar onde xx se refere ao número do grupo (consultar “grupos por turma” na página Moodle da disciplina).

---

Este trabalho explora os princípios básicos da norma JPEG (modo sequencial) para compressão de imagens com perdas. Neste trabalho apenas serão exploradas as imagens em níveis de cinzento.

1. Construa uma função (codificador) que para cada bloco de  $8 \times 8$  da imagem original efetue a DCT bidimensional e construa outra função (descodificador) que faça a DCT inversa. Ver funções: `cv2.dct()` e `cv2.idct()`  
Visualize a imagem depois da DCT e verifique que a DCT inversa resulta na imagem original.
2. Construa uma função (codificador) que para cada bloco de  $8 \times 8$  de coeficientes da transformação efectuada faça a divisão pela matriz de quantificação (tabela K1 no anexo da norma) multiplicada por um factor de qualidade  $q$  (ver função `quality_factor` do ficheiro `Tabelas_jpeg.ipynb`). Veja a imagem com o conjunto dos blocos após a quantificação. Construa uma função (descodificador) que realize a operação inversa da quantificação. Junte estas funções às já realizadas e verifique para diferentes fatores de qualidade qual a SNR e veja a imagem descodificada.
3. Construa uma função (codificador) que faça a codificação diferencial dos coeficientes DC após a quantificação. Construa a função inversa para o descodificador.
4. Construa uma função (codificador) que crie um array com a indexação em zig-zag dos coeficientes AC após a quantificação e crie um array com os pares (zero run length, nonzero value). Construa a função inversa para o descodificador.
5. Junte estas funções às já realizadas e veja a imagem descodificada.
6. Construa uma função que dados os arrays das alíneas anteriores use as tabelas do código de Huffman (tabela K3 e K5) e grave num ficheiro a sequência de bits correspondente. (não é necessário usar o formato JFIF)
7. Construa uma função que leia o ficheiro gravado e retorne os arrays com os coeficientes AC e DC.
8. Junte estas funções às já realizadas e veja a imagem descodificada.  
Para diferentes factores de qualidade meça a relação sinal-ruído e a taxa de compressão obtida. Represente um gráfico onde se apresente a taxa de compressão em função do SNR.
9. No mesmo gráfico compare o seu compressor de imagem com outros existentes para várias qualidades.
10. O Jupyter Notebook o deve conter uma descrição breve das funções realizadas e uma tabela com todos os resultados da SNR, taxa de compressão, tempo de compressão e descompressão.