scc0251/5830— Prof. Moacir Ponti

PAE/monitores: Fernando Pereira dos Santos, Eduardo Carlos Souza

Trabalho 7: descrição de imagens

Desenvolva seu código sem olhar o de colegas. Plágio não será tolerado.

Nesse trabalho você deverá implementar um método de descrição de imagem utilizando as informações de: cor, textura e gradiente da imagem. Esse método será então aplicado para tentar encontrar um determinado objeto numa imagem de maior resolução.

Você deverá implementar usando `python 3` com as bibliotecas `numpy` e `imageio`.

Leia com calma as instruções de cada etapa.

1. **Receber via teclado**:
   a) nome do arquivo da imagem com objeto de interesse $o(x, y)$,
   b) nome do arquivo da imagem para realizar busca $g(x, y)$,
   c) parâmetro de quantização em bits b.

As imagens serão lidas no formato png.

Descrição de imagens

Assumindo que a imagem esteja representada corretamente, é possível considerar a extração de atributos visuais que permitam descrever a cena ou objeto por meio de características. Podemos ver esse processo como um mapeamento da matriz de pixels imagem X em um espaço com m dimensões, ou seja, uma função $f : X \to \mathbb{R}^m$.

Para que seja possível comparar imagens, definimos distâncias entre vetores de características $D(a, b)$, sendo a e b produzidos pela mesma função f. Ao final do processo encontramos uma imersão comumente num espaço de Hilbert, na qual as imagens são representadas por vetores. De forma geral, as funções de distância mais utilizadas são as de Minkowski:

$$D_p(a, b) = \left[\sum_{i=1}^{m} (a_i - b_i)^p \right]^{1/p}. \quad (1)$$

Comumente a distância Euclidiana (Minkowski para o caso $p = 2$), é empregada. A depender do cenário, variações podem ser utilizadas, como ponderar a soma das diferenças.

Nas seções a seguir descreveremos três grupos de características visuais, em particular Cor, Textura e Gradiente. Para isso é preciso empregar um método de conversão de RGB para um espaço quantizado com um único canal de cor (tons de cinza) de 8 bits ou menos. Utilizaremos o método "Luminance", computado em termos dos canais RGB como:

$$f(x, y) = \lfloor 0.299R + 0.587G + 0.114B \rfloor, \quad (2)$$

em que $\lfloor . \rfloor$ é um arredondamento para baixo.

Após computar a imagem f, definiremos um parâmetro b como a quantidade de bits a serem considerados. Por exemplo se $b = 6$, então você deverá quantizar a imagem em 64 intensidades. Faça isso por meio do deslocamento de bits (bit shift) em $8 - b$ posições.

Histograma de Cor

Esse processo visa estimar a frequência com que cada cor aparece na imagem. Pode ser vista como uma função $h(k)$, sendo $k \in [0, 255]$ para imagens com 256 intensidades de cor. Se normalizarmos $h(k)$ de forma que a soma de todos seus valores avaliados para todo k seja 1, então temos a densidade:

$$p(k) = \frac{h(k)}{\sum_{k=0}^{C-1} h(k)}, \tag{3}$$

que pode ser vista como a probabilidade da ocorrência da cor k ao selecionarmos um pixel aleatoriamente, e C é a quantidade de cores/amplitudes possíveis na imagem (definida pelo parâmetro b descrito anteriormente). Essa densidade será o vetor de características de cor $\mathbf{d}_c$.

Descritores de Textura de Haralick

Definimos uma matriz de A de tamanho $C \times C$, na qual cada célula na posição $a_{i,j}$ indica a frequência com a qual pixels p de intensidade k_i ocorrem relativo a pixels q com intensidade k_j, sendo q definido por meio de um operador de deslocamento $Q(.)$ com relação a p, i.e. $q = Q(p)$. Seja $Q(p) = p + (1, 1)$, ou seja deslocamento para baixo e à direita, e uma imagem com 3 tons de cinza, $k_1 = 0, k_2 = 1, k_3 = 2$. Desconsiderando os pixels da borda na última linha e coluna (para os quais não existe q), teremos uma matriz A de tamanho 3×3

1	1	2	2	2x
1	1	1	2	2x
1	1	0	2	2x
0	0	2	0	2x
1x	2x	2x	2x	2x

	0	1	2
0	1	0	3
1	1	3	2
2	0	0	5

	0	1	2
0	1/15	0	1/5
1	1/15	1/5	2/15
2	0	0	1/3

Ao normalizar A para que sua soma seja unitária, obtemos a matriz de co-ocorrência G que codifica as probabilidades de um pixel escolhido ao acaso possuir a relação de co-ocorrência $g_{i,j}$, codificando assim uma função densidade probabilidade conjunta para um par de cores.

Medidas computados sobre essa matriz são úteis para descrever textura. Esses são conhecidos por descritores de Haralick. Sejam as médias e os desvios padrão nas direções

das linhas i e colunas j da matriz dados por:

$$\mu_i = \sum_{i=0}^{C-1} i \sum_{j=0}^{C-1} G(i,j) \;\; \mu_j = \sum_{j=0}^{C-1} j \sum_{i=0}^{C-1} G(i,j)$$
$$\sigma_i = \sum_{i=0}^{C-1} (i-\mu_i)^2 \sum_{j=0}^{C-1} G(i,j) \;\; \sigma_i = \sum_{j=0}^{C-1} (j-\mu_j)^2 \sum_{i=0}^{C-1} G(i,j)$$

Os descritores a serem computados são

1. Energia:
$$\sum_{i=0}^{C-1} \sum_{j=0}^{C-1} G(i,j)^2 \tag{4}$$

2. Entropia:
$$-\sum_{i=0}^{C-1} \sum_{j=0}^{C-1} G(i,j) \log[G(i,j)+\varepsilon], \tag{5}$$
em que $\varepsilon = 0.001$ é um valor pequeno para evitar $\log(0)$.

3. Contraste:
$$\frac{1}{(C-1)^2} \sum_{i=0}^{C-1} \sum_{j=0}^{C-1} (i-j)^2 G(i,j) \tag{6}$$

4. Correlação:
$$\frac{\sum_{i=0}^{C-1} \sum_{j=0}^{C-1} ij\,G(i,j) - \mu_i \mu_j}{\sigma_i \sigma_j}, \tag{7}$$
o qual requer $\sigma_i, \sigma_j > 0$, do contrário a correlação é dada como nula (0).

5. Homogeneidade:
$$\sum_{i=0}^{C-1} \sum_{j=0}^{C-1} \frac{G(i,j)}{1+|i-j|} \tag{8}$$

Você deverá computar esses 5 descritores para uma matriz obtida com deslocamento $Q(1,1)$ na imagem, desconsiderando os pixels para os quais não seja possível analisar esse deslocamento. Esses compõem o vetor descritor de textura $\mathbf{d}_t$.

Descritor de Orientações do Gradiente

Computar sobre a imagem quantizada o gradiente da imagem nas direções x e y, i.e. g_x e g_y utilizando os operadores de Sobel horizontal e vertical:

$$w_{Sx} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \; ; \; w_{Sy} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{9}$$

Ignorando os pixels da borda da imagem, computar a magnitude e o ângulo do gradiente:

$$M(x,y) = \frac{\sqrt{g_x^2 + g_y^2}}{||\sqrt{g_x^2 + g_y^2}||},$$
$$\phi(x,y) = \tan^{-1}\frac{g_y}{g_x}.$$

, em que $||\sqrt{g_x^2 + g_y^2}||$ é a soma dos pixels da magnitude e tem o objetivo de normalizar M de forma que sua soma seja unitária.

Finalmente, a partir desses computamos o histograma dos ângulos. Quantizar os ângulos em 18 intervalos, ou seja com bins de tamanho 20, considerando os intervalos 0-19, 20-39, ..., 340-359. Ao invés de contar de forma discreta cada ângulo, você deverá ao invés somar no respectivo bin a magnitude $M(x,y)$. Esses compõem o vetor descritor de gradiente $\mathbf{d}_g$

Tarefa

Seu programa deverá aplicar a conversão para tons de cinza e a quantização nas imagens $o(x,y)$ e $g(x,y)$. A imagem o terá sempre tamanho 32×32 pixels, e a imagem g sempre tamanho 256×256 pixels.

Após isso, para a imagem $o(x,y)$, computar os descritores de cor, textura e gradiente conforme definidos anteriormente, e concatená-los. O vetor final terá tamanho igual a $2^b + 5 + 18$, sendo os 2^b primeiros relativos à cor, os 5 seguintes à textura e os 18 últimos relativos ao gradiente. O vetor da imagem o é $\mathbf{d}_o$

Para a imagem $g(x,y)$, você deverá computar o descritor para janelas deslizantes de tamanho 32×32 e com passo 16, ou seja, com sobreposição de 50% entre as janelas. O vetor de cada janela j é $\mathbf{d}_j$. A janela $j = 1$ define os pixels nos intervalos $x = [0, 31], y = [0, 31]$, a janela $j = 2$ os pixels nos intervalos $x = [0, 31], y = [15, 47]$ a janela $j = 3$ os pixels $x = [0, 31], y = [31, 63]$, e assim por diante, seguindo deslocando em y para cada intervalo fixo em x.

Você deverá, para cada janela computar os descritores da mesma forma como foi feito para a imagem de entrada o, e computar a distância entre $\mathbf{d}_o$ e cada $\mathbf{d}_j$,

$$D_2(\mathbf{d}_o, \mathbf{d}_j) = \left[\sum_{i=1}^{m} \mathbf{w}[i](\mathbf{d}_o[i] - \mathbf{d}_j[i])^2\right]^{1/2}, \tag{10}$$

onde $\mathbf{w}$ é um vetor de pesos do mesmo tamanho dos vetores de características, deve ser definido de forma que os descritores de textura, cor e gradiente tenham o mesmo peso na distância (do contrário por exemplo o gradiente terá mais peso do que a textura pois tem 18 elementos contra 5 da textura).

Seu programa deverá retornar o número da janela relativa a menor distância, i.e. o

valor de j para o qual:

$$\arg\min_j D_2(\mathbf{d}_o, \mathbf{d}_j) \tag{11}$$

Esquema com os números das janelas de busca na imagem:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100	101	102	103	104	105
106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165
166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
181	182	183	184	185	186	187	188	189	190	191	192	193	194	195
196	197	198	199	200	201	202	203	204	205	206	207	208	209	210
211	212	213	214	215	216	217	218	219	220	221	222	223	224	225

Entrada e saída

Exemplo de entrada (quantização em 6 bits, $2^6 = 64$ tons de cinza):

```
objeto.png
imagem.png
6
```

Exemplos de saída: apenas o número da janela com a menor distância.

```
5
```

Submissão e instruções

No sistema Run.Codes deve incluir apenas o arquivo `.py`

1. **É obrigatório comentar seu código**. Como cabeçalho insira seu nome, número USP, código da disciplina, ano/semestre e o título do trabalho. Haverá desconto na nota caso esse cabeçalho esteja faltando, além de descontos para falta de comentários.

2. **É obrigatório organizar seu código em funções**.