

Questão 1

Incompleto

Vale 10,00 ponto(s).

Análise dos dados da base SceneAttribute

Contexto

Visão Computacional é área de pesquisa que trata do desenvolvimento de algoritmos para extrair informações a partir de dados visuais (imagens ou vídeos). Um dos desafios nessa área é o seguinte: *dada uma imagem, indique quais objetos estão presentes nela e indique atributos importantes desses objetos, tais como a cor e a posição deles na imagem*. Atualmente, a abordagem mais comum para esse problema se baseia no uso de técnicas de aprendizado de máquina (uma sub-área de inteligência artificial). Para se projetar tais algoritmos e avaliar seus resultados, é necessário que haja um conjunto de dados contendo imagens e seus respectivos rótulos de anotação.

Um desses conjuntos de dados é o [Scene Attribute Dataset](#), criadas por pesquisadores da Universidade de Pequim e da Universidade de California (Los Angeles). Trata-se de um conjunto de milhares de imagens e alguns arquivos auxiliares. Um desses arquivos contém um texto que indica, para cada arquivo de imagem e para cada par atributo-objeto, quais são as coordenadas do retângulo (*bounding box*) onde esse atributo-objeto aparece na imagem.

Objetivo

Desenvolva um programa em Python3 que processe os dados textuais da base [Scene Attribute](#) de forma a facilitar a análise desses dados. Tal análise servirá como um ponto inicial para algoritmos de aprendizado de máquina (a serem implementados para os projetos posteriores).

Entrada

Os dados a serem fornecidos tem o formato abaixo:


```
T N  
  
nome_do_arquivo_1  
atributo-objeto_1  
x1 y1 x2 y2  
  
nome_do_arquivo_2  
atributo-objeto_2  
x1 y1 x2 y2  
  
.  
.  
.  
nome_do_arquivo_N  
atributo-objeto_N  
x1 y1 x2 y2
```

onde:

- **T** é um número (int) que indica a tarefa a ser desempenhada pelo seu programa (detalhes na [próxima seção](#)).
- **N** indica o número (int) de blocos de dado a serem listados.
- **nome_do_arquivo_1** é uma palavra (string) que contem o nome de um arquivo de imagem.
- **atributo-objeto_1** é uma palavra que indica um par atributo-objeto que aparece na imagem.
- **x1** é a primeira coordenada horizontal (coluna) do retângulo que contem o objeto na imagem.
- **y1** é a primeira coordenada vertical (linha) do retângulo que contem o objeto na imagem.
- **x2** é a última coordenada horizontal (coluna) do retângulo que contem o objeto na imagem.
- **y2** é a última coordenada vertical (linha) do retângulo que contem o objeto na imagem.

Há um bloco de dados para cada atributo-objeto que aparece em uma imagem, cada bloco é separado por uma linha em branco. Quando uma imagem contem múltiplos objetos, há múltiplos blocos de dados com o mesmo nome de arquivo.

Todos os números fornecidos na entrada são naturais (inteiros não negativos).

Segue a lista de possíveis pares atributo-objeto:

1. black-bison
2. elephant
3. white-horse
4. brown-horse
5. scarlet-ibis
6. black-ibis
7. white-ibis
8. blue-sky
9. overcast-sky
10. cloudy-sky
11. dusthaze-sky
12. rocky-mountain
13. snowy-mountain
14. birdseye-building
15. perspective-building
16. front-building
17. red-flower
18. purple-flower
19. pink-flower
20. sand
21. tree
22. green-field



- 23. snowy-field
- 24. yellow-field
- 25. road
- 26. tower
- 27. blue-ocean
- 28. green-cliff
- 29. black-cliff
- 30. waterfall

Sobre as coordenadas dos retângulos nas imagens, é usado o padrão de coordenadas em pixels, em que o primeiro elemento fica no canto **superior** esquerdo da imagem, índices de colunas (x) crescem para a direita e índices de linhas (y) crescem para **baixo**.

Exemplo

A Figura a seguir é uma das milhares de imagens disponíveis no conjunto de dados. Ela mostra um bisão (búfalo-americano) na neve. As imagens tem 256 x 256 pixels. O ponto mais superior a esquerda tem coordenadas (0,0). O ponto mais inferior a esquerda tem coordenadas (0,256). O ponto mais superior a direita tem coordenadas (256,0) e o ponto mais inferior a direita tem coordenadas (256, 256). Portanto, cada pixel é um intervalo entre 4 coordenadas discretas.



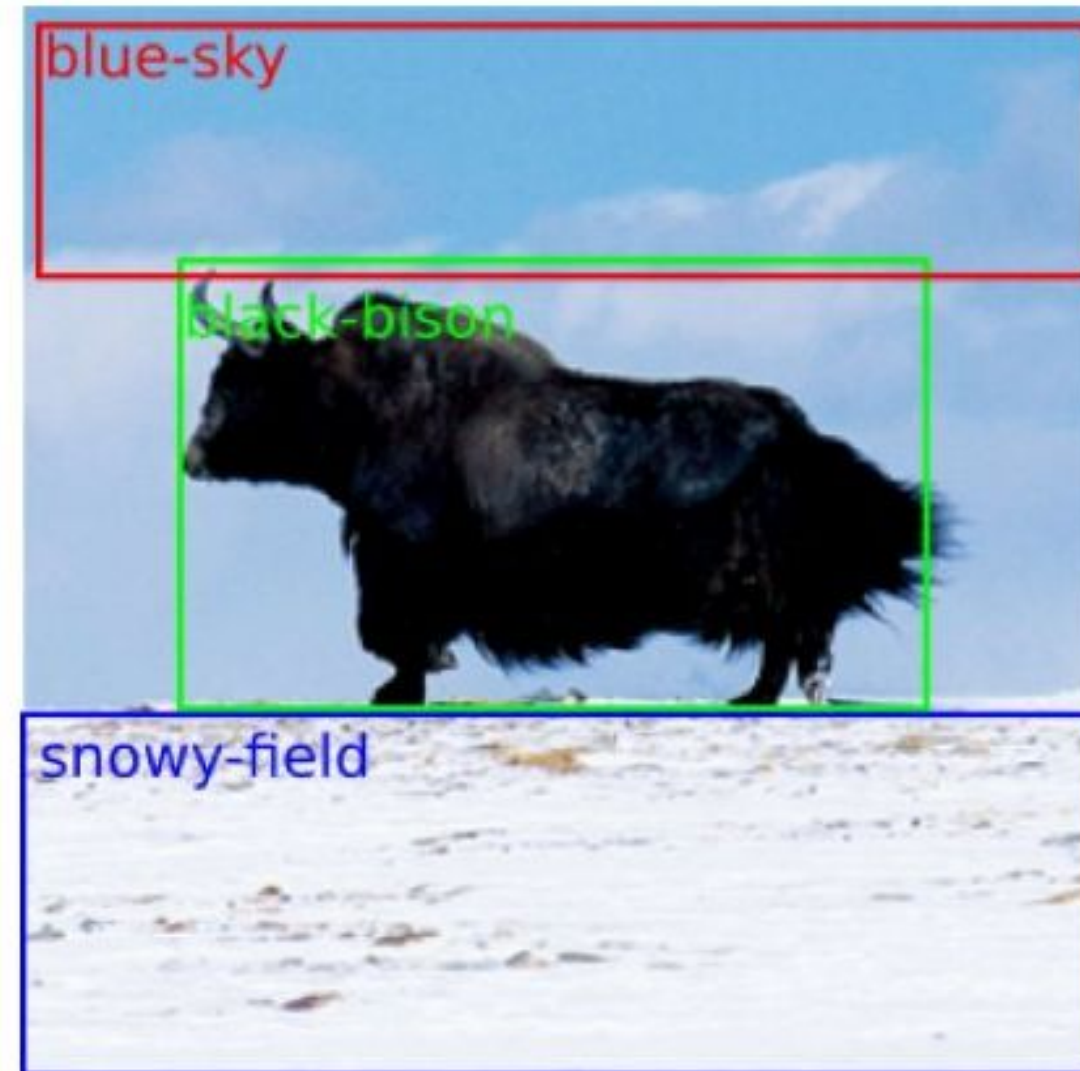
Suponto a seguinte descrição para a Figura anterior: O `nome_do_arquivo` é `bison_022` e nele temos três `atributo-objeto` com suas respectivas coordenadas. O primeiro objeto na imagem é o próprio bisão centrado, o segundo objeto é o céu azul na parte superior da imagem e o terceiro é um campo com neve na parte inferior da imagem.

```
bison_022  
black-bison  
52 68 208 163
```

```
bison_022  
blue-sky  
9 6 253 78
```

```
bison_022  
snowy-field  
4 168 254 254
```

A figura a seguir ilustra tais coordenadas:



Tarefas e Saída

Como dito acima, o valor do primeiro número fornecido na entrada indica qual tarefa deve ser desempenhada pelo seu programa. A seguir há uma descrição de cada tarefa.

1. Impressão em formato CSV (T=1)

Imprima na saída padrão um texto com N linhas, sendo que cada linha possui 6 colunas, separadas por vírgula, com o seguinte conteúdo

```
nome_do_arquivo_1,atributo-objeto_1,x1,y1,x2,y2
nome_do_arquivo_2,atributo-objeto_2,x1,y1,x2,y2
.
.
.
nome_do_arquivo_N,atributo-objeto_N,x1,y2,x2,y2
```

2. Percentual de atributos-objetos (T=2)

Conte quantas vezes cada atributo-objeto aparece nos dados de entrada e imprima o percentual.

Imprima a saída de dados neste formato:


```
black-bison: X.X  
elephant: X.X  
white-horse: X.X  
.  
.  
.  
waterfall: X.X
```

onde X.X é um número em ponto flutuante com uma casa decimal com o percentual de blocos que contem o respectivo atributo-objeto. Isso é calculado por 100 vezes a soma do número de blocos contendo cada atributo dividida por N.

3. Média das coordenadas e dos tamanhos (T=3)

Imprima a coordenada (x,y) média da posição central de todos os objetos, bem como a largura e a altura média deles. O valor impresso deve ser um arredondamento para inteiro (e não truncamento). Os dados da saída devem ser apresentados como 4 números inteiros, desta forma:

```
x_medio y_medio largura_media altura_media
```

4. Maiores, menores e suas posições (T=4)

Sabendo-se que as coordenadas dos pixels nas imagens vão sempre de 0 a 256 (tanto para linhas quanto para colunas) e que o centro das imagens fica na coordenada (128,128). Assuma que a posição de cada atributo-objeto é definida através da média entre as coordenadas x e y de sua caixa. Identifique, dentre todas imagens, qual dos atributos-objetos

- fica mais próximo do centro da imagem;
- fica mais à esquerda na imagem;
- fica mais à direita na imagem;
- fica mais acima na imagem;
- fica mais abaixo na imagem;
- tem a maior área;
- tem a menor área;

No caso de empate em qualquer dos itens acima, o primeiro é o que deve constar na saída do seu programa.

Imprima seus resultados da seguinte forma, identificando o atributo-objeto e a imagem:

```
mais central: atributo-objeto,nome_do_arquivo
mais a esquerda: atributo-objeto,nome_do_arquivo
mais a direita: atributo-objeto,nome_do_arquivo
mais acima: atributo-objeto,nome_do_arquivo
mais abaixo: atributo-objeto,nome_do_arquivo
maior area: atributo-objeto,nome_do_arquivo
menor area: atributo-objeto,nome_do_arquivo
```

5. Busca Booleana por imagens (T=5)

Assuma que os dados de entrada são sempre fornecidos de maneira que blocos referentes à mesma imagem aparecem consecutivos. Por exemplo, se há 2 imagens, sendo que a primeira contem 3 objetos e a segunda contem outros 4 objetos, serão dados 7 blocos de forma que os 3 primeiros blocos necessariamente contem dados sobre a primeira imagem e os últimos 4 blocos necessariamente contém dados da segunda imagem. Indique quais imagens possuem **árvore** (*tree*) mas **não** possuem **nenhum tipo de campo** (*green-field*, *snowy-field* nem *yellow-field*).

Formato da saída:

Quando forem encontradas M imagens com as especificações acima, imprima o nome de cada arquivo:

```
nome_do_arquivo_1
nome_do_arquivo_2
.
.
.
nome_do_arquivo_M
```

Caso contrário, imprima

```
nada
```

Observação

Sempre que for executada uma sequência de cálculos, execute-a de maneira a minimizar a propagação de erros. Por exemplo, para se calcular a média entre as variáveis A, B e C, é melhor calcular desta forma: $(A+B+C)/3$, e **não** desta: $A/3 + B/3 + C/3$. Apesar de serem matematicamente equivalentes, a segunda forma é numericamente mais incerta e deve ser evitada.

For example:

Input	Result
1 8 horse_040 brown-horse 63 82 244 194 horse_040 snowy-field 11 209 252 250 horse_040 dusthaze-sky 5 3 175 85 horse_071 green-field 11 192 259 250 horse_071 brown-horse 69 38 220 200 horse_071 tree 112 10 245 73 mountain_033 snowy-mountain 9 75 261 221 mountain_033 blue-sky 7 2 248 81	horse_040,brown-horse,63,82,244,194 horse_040,snowy-field,11,209,252,250 horse_040,dusthaze-sky,5,3,175,85 horse_071,green-field,11,192,259,250 horse_071,brown-horse,69,38,220,200 horse_071,tree,112,10,245,73 mountain_033,snowy-mountain,9,75,261,221 mountain_033,blue-sky,7,2,248,81
2 14 highway_art1674 blue-sky 9 4 248 134	black-bison: 7.1 elephant: 0.0 white-horse: 7.1 brown-horse: 7.1 scarlet-ibis: 0.0 black-ibis: 0.0

Input	Result
highway_art1674 road 19 156 250 251	white-ibis: 0.0 blue-sky: 14.3 overcast-sky: 0.0 cloudy-sky: 0.0
horse_045 white-horse 24 77 217 215	dusthaze-sky: 21.4 rocky-mountain: 0.0 snowy-mountain: 0.0 birdseye-building: 0.0
horse_045 green-field 7 193 252 254	perspective-building: 0.0 front-building: 0.0 red-flower: 0.0 purple-flower: 0.0
horse_045 tree 9 29 157 81	pink-flower: 0.0 sand: 0.0 tree: 7.1 green-field: 14.3
horse_045 dusthaze-sky 7 4 190 27	snowy-field: 7.1 yellow-field: 7.1 road: 7.1 tower: 0.0
horse_059 brown-horse 34 28 210 216	blue-ocean: 0.0 green-cliff: 0.0 black-cliff: 0.0 waterfall: 0.0
horse_059 yellow-field 8 220 255 254	
bison12 black-bison 15 85 146 167	
bison12 green-field 12 164 248 253	
bison12 dusthaze-sky 14 8 252 92	

Input	Result
bison12 dusthaze-sky 141 6 249 173 opencountry_161 blue-sky 16 6 243 62 opencountry_161 snowy-field 9 59 255 251	
3 15 tower02 overcast-sky 17 5 188 197 tower02 tower 202 7 247 197 tower02 blue-ocean 6 208 253 252 palace_001 dusthaze-sky 9 5 247 60 palace_001 front-building 7 72 253 228 palace_001 yellow-field 9 221 256 257 field34 tree 14 8 247 120	136 127 191 111

Input	Result
field34 green-field 8 142 250 250	
street_par85 dusthaze-sky 102 1 162 122	
street_par85 perspective-building 163 3 251 173	
street_par85 perspective-building 7 2 119 157	
street_par85 road 34 155 251 249	
mountain_009 cloudy-sky 10 3 254 50	
mountain_009 rocky-mountain 17 63 242 172	
mountain_009 green-field 5 182 249 257	
5 15	nada
ibis_004 white-ibis 30 103 218 205	
ibis_004 blue-ocean	

Input	Result
3 5 210 90 ibis_004 blue-ocean 18 198 242 254 mountain_054 blue-sky 11 5 249 52 mountain_054 snowy-mountain 8 56 256 244 ibis_032 black-ibis 85 6 245 199 ibis_032 green-field 3 73 112 248 ibis_032 green-field 131 5 250 72 ibis_032 green-field 9 196 255 256 street_par177 perspective-building 151 30 261 170 street_par177 perspective-building 0 0 108 159 street_par177 road	

Input	Result
75 158 252 252 opencountry_172 yellow-field 6 171 250 252 opencountry_172 tree 7 123 255 169 opencountry_172 blue-sky 9 1 253 127	
5 28 waterfall_074 black-cliff 18 10 153 272 waterfall_074 waterfall 161 8 212 275 waterfall_074 black-cliff 218 4 261 244 opencountry_test_010 red-flower 14 106 247 257 opencountry_test_010 overcast-sky 3 0 248 84 highway_bost183 cloudy-sky 5 8 233 108 highway_bost183	highway_bost183

Input	Result
tree 3 116 105 162	
highway_bost183 tree 192 104 256 157	
highway_bost183 road 6 169 254 252	
opencountry_076 cloudy-sky 4 2 247 137	
opencountry_076 yellow-field 6 132 252 231	
opencountry_092 overcast-sky 8 7 248 130	
opencountry_092 tree 17 128 213 160	
opencountry_092 yellow-field 6 169 250 220	
opencountry_092 green-field 7 225 251 251	
mountain_086 dusthaze-sky 7 6 251 57	
mountain_086	

Input	Result
rocky-mountain 6 60 253 252	
ibis_001 black-ibis 52 67 228 198	
ibis_001 green-field 75 10 242 111	
ibis_001 green-field 12 180 247 238	
bison08 tree 1 3 251 93	
bison08 black-bison 18 95 242 207	
bison08 green-field 4 194 251 251	
volcano_0191 dusthaze-sky 9 6 256 126	
volcano_0191 rocky-mountain 9 124 247 255	
horse_097 tree 15 5 147 76	
horse_097	

Input	Result
white-horse 30 47 230 194 horse_097 green-field 0 165 251 249	
4 6 highway_gre458 overcast-sky 4 3 252 97 highway_gre458 road 19 149 245 252 opencountry_066 sand 7 153 254 253 opencountry_066 cloudy-sky 7 6 247 149 mountain_037 blue-sky 10 13 256 99 mountain_037 snowy-mountain 18 107 256 248	mais central: snowy-mountain,mountain_037 mais a esquerda: cloudy-sky,opencountry_066 mais a direita: snowy-mountain,mountain_037 mais acima: overcast-sky,highway_gre458 mais abaixo: sand,opencountry_066 maior area: cloudy-sky,opencountry_066 menor area: blue-sky,mountain_037

Input	Result
4 2 ocean51 cloudy-sky 7 10 244 97 ocean51 blue-ocean 19 133 242 240	mais central: blue-ocean,ocean51 mais a esquerda: cloudy-sky,ocean51 mais a direita: blue-ocean,ocean51 mais acima: cloudy-sky,ocean51 mais abaixo: blue-ocean,ocean51 maior area: blue-ocean,ocean51 menor area: cloudy-sky,ocean51

Answer: (penalty regime: 0, 0, 10, 20, ... %)

1 |

Verificar