****

**CS4186: Computer Vision and Image**

**Sem B 2021-22**

**Assignment 1 Report**

**Topic: Instance Search**

Date: 23 April 2022

|  |  |
| --- | --- |
| Student Name | Student ID |
| Rahil Raj Mohan | 55414617 |

**Introduction**

The first assignment centred on implementing a multitude of methods for instance search by matching images present in a query folder (number=20) to instances in a gallery dataset (number=5000). To this extent, 2 methods were selected and implemented. The first is an approach which generates **Colour Histograms** of the query image set and gallery image set and compares them using a distance metric. The second is via a **Convolutional Neural Net** feature extractor to extract associated complex features from each gallery instance and similarly compare it to the selected query image.

Three distances measures - Cosine similarity, Euclidean distance and Pearson’s Coefficient, will be used to find the closest matches for the query images.

**Dataset**

The dataset is broadly divided into 2 folders – ‘query\_4186’ and ‘gallery\_4186’.

A third folder called ‘query\_txt\_4186’ contains bounding boxes for the query images. Thus, the first task is to crop the query images as per the coordinates present in the text files under the given folder. This is carried out by the ‘query\_crop()’ function defined under the ‘utils.py’ script.

‘utils.py’ contains several useful utility functions and adds a layer of abstraction as well as the ability to reuse functions. Taking the example of one such query image, we have –

A picture containing indoor

Description automatically generatedA picture containing blue, indoor, decorated, close

Description automatically generated

using ‘316.txt’

Figure 1:Image - 316.jpg (post-cropped)

Figure 2:Image - 316.jpg (pre-cropped)

All 20 query images are then placed under a new folder titled ‘query\_cropped’ and will be used for instance matching.

**Algorithms**

1. **Colour Histogram**

This a computationally inexpensive method that generates the colour histogram of the selected query and gallery images and compares the associated colour spaces. Though simple, it is not consistent as minor variations in orientation, lighting, noise, discolouration due to reflections can drastically decrease performance. Plus, this method cannot identify the shape or contours of the objects in the images. This method was self-implemented with 16 bins with bin size of 16 each.

The major steps are:

1. Crop Images
2. Generate colour histograms from the query images and gallery images via the ‘extract\_color\_histogram()’ function, which are then subsequently stored under ‘query\_cropped\_hist’ and ‘gallery\_feature\_hist’ folders respectively in the ‘.npy’ format.
3. Iterate through the ‘query\_cropped\_hist’ folder and extract the relevant feature file for each image. These features are then contrasted to each gallery image feature file present in ‘gallery\_feature\_hist’.
4. Sort the rank matrix for each query image and return the top 10 most similar gallery images.
5. Visualise the results and note them down.

With Euclidean Distance,

Rank List =

Q1: 3141 4893 1210 4992 775 3 4223 4845 2894 591

Q2: 1871 2533 4461 3884 2781 4353 2490 4393 1759 836

Q3: 291 4214 24 2215 376 3154 1328 2661 3857 4122

Q4: 2285 892 770 4148 1946 297 3398 1048 3335 3742

Q5: 3156 248 1657 1474 3219 1823 1755 1818 603 4676

Q6: 3889 1031 3379 4590 4218 4583 2298 1872 4462 4075

Q7: 3259 819 2503 4996 1470 1547 2272 3161 3065 484

Q8: 2089 949 4749 591 3503 4527 4623 2419 2324 3211

Q9: 3219 453 792 1409 219 1004 2022 1474 2552 1506

Q10: 3860 3447 3231 3550 2183 3323 4686 253 2295 117

Q11: 1518 4991 2254 579 2976 3014 1226 164 416 1756

Q12: 3543 3485 722 573 1353 3151 3578 3762 1052 211

Q13: 653 2098 2593 3676 752 1625 1291 97 949 4140

Q14: 1038 1682 2033 2194 1919 3514 1095 4700 1486 4167

Q15: 1662 2669 822 2023 1515 854 4679 2369 1533 3332

Q16: 2 1906 3029 3306 4403 1412 2529 3727 1987 3622

Q17: 1419 3800 5007 4419 4992 2006 3084 2134 166 232

Q18: 3168 4040 1881 1517 1180 972 2117 2498 2749 4320

Q19: 3862 4695 694 3688 4441 2348 4348 624 611 4904

Q20: 2921 3770 1213 2173 272 769 1481 4099 437 4349

With Pearson Distance,

Rank List =

Q1: 3141 1860 775 1210 2403 4992 4893 3 766 4223

Q2: 2533 2901 3955 662 4171 2988 2781 1377 977 2749

Q3: 291 4214 376 3926 3857 65 24 3005 3154 4331

Q4: 2285 3090 770 892 2199 2156 845 3950 3716 4850

Q5: 2850 2666 4280 1621 2995 3766 1186 45 958 4098

Q6: 1031 3379 3889 4590 421 4029 4218 4583 4462 1872

Q7: 3259 2272 819 1547 2503 4996 1470 3161 3065 280

Q8: 2089 949 4256 4527 2653 3827 200 4749 86 1130

Q9: 1186 4262 399 4493 1301 2900 2458 1474 2039 4550

Q10: 3206 1936 1994 414 1149 1420 181 4506 2295 4677

Q11: 416 4991 164 2976 1518 2254 3014 579 2465 1226

Q12: 161 687 1026 1127 4046 3726 4 510 2991 212

Q13: 653 2098 2593 1625 3676 752 86 2008 4140 200

Q14: 1682 1038 2033 3917 4120 1838 1919 4774 1701 2194

Q15: 2669 1662 2834 3332 4679 1221 2369 3579 854 2096

Q16: 2 1987 3114 3622 2168 1808 1331 4835 1412 2245

Q17: 1419 1525 2006 4796 1266 4937 5007 3800 5017 2224

Q18: 3168 1363 4040 4036 4437 3413 1180 2709 1517 2117

Q19: 3862 694 4695 3688 1847 2563 4348 2680 611 4904

Q20: 769 2315 233 272 3770 2690 1213 2173 2921 170

With Cosine Distance,

Rank List =

Q1: 3141 1210 775 4893 4992 3 4223 4845 766 2894

Q2: 1871 2533 2781 4461 3884 4393 2490 2468 4353 1759

Q3: 291 4214 376 2215 3857 3154 3627 4009 3926 2661

Q4: 2285 892 770 4148 4850 3398 1946 3335 297 1889

Q5: 2850 3156 248 1621 1657 1474 4280 2666 3219 1818

Q6: 1031 3379 3889 4590 421 4583 4029 4218 4462 1872

Q7: 3259 2272 819 2503 1547 4996 1470 3161 3065 484

Q8: 949 2089 4749 4527 3503 591 2653 4623 3211 2419

Q9: 3219 1409 1474 1186 219 2022 792 1004 4262 2458

Q10: 3860 3550 3231 2183 4300 4686 3447 3183 117 1817

Q11: 416 1518 4991 2254 579 164 2976 1226 4313 1756

Q12: 3543 722 573 3485 1353 1026 3151 3578 4046 3762

Q13: 653 2098 2593 3676 752 1625 97 1291 4140 949

Q14: 1682 1038 2033 2194 1919 1095 3514 4700 1486 3008

Q15: 2669 1662 4679 854 2369 3332 822 2023 1515 1221

Q16: 2 1987 3622 1412 2168 4835 4399 1331 4403 4886

Q17: 1419 2982 2134 3084 2224 4992 166 3800 2058 2006

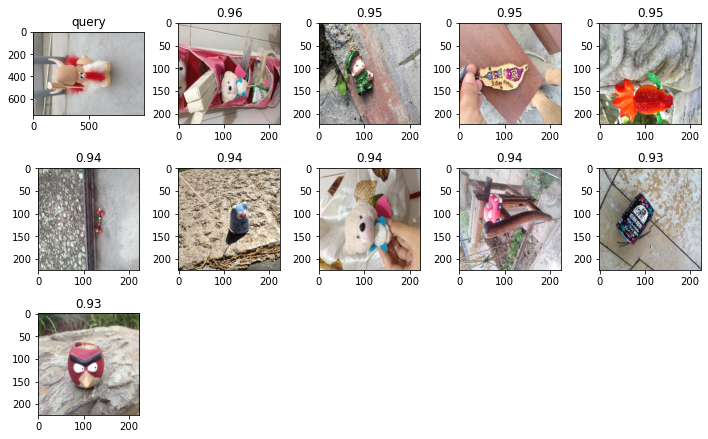
Q18: 3168 4040 1180 1517 1363 1881 2498 4320 2117 972

Q19: 3862 4695 3688 4348 4441 2563 624 611 4282 694

Q20: 1213 3770 272 2173 2921 769 2690 233 2315 170

To analyse the results, we display the top 10 images for Q1, Q3, Q5 obtained using Cosine.

Q1:



Q3:

A picture containing text, screen, several

Description automatically generated

Q5:

A picture containing text

Description automatically generated

We can clearly observe that images are being matched purely on the colour space similarity rather than object similarity. This is particularly noticeable in Q3 where blue images (like the blue poster, blue door) were noted to have high similarity despite being mismatched with the query image (a blue card).

1. **CNN (VGG-19)**

Convolutional Neural Networks play a crucial role in extracting deep features and spatial information from images. This project uses the convolutional layers of the VGG-19

With Euclidean Distance,

rankList =

Images,

With Pearson Distance,

rankList =

Images,

With Cosine Distance,

rankList =

Images,

At later stages, we will implement a sliding window to extract subsections of the gallery image and try to find instances of any query image within these subsections.

1. CNN (Xception Algorithm with Sliding Window)