CENG 483

Introduction to Computer Vision

Fall 2021-2022

Take Home Exam 1
Instance Recognition with Color Histograms
Student ID:

Please fill in the sections below only with the requested information. If you have additional things you want to mention, you can use the last section. For all of the configurations make sure that your quantization interval is divisible by 256 in order to obtain equal bins.

1 3D Color Histogram

In this section, give your results without dividing the images into grids. Your histogram must have at most 4096 bins. E.g. Assume that you choose 16 for quantization interval then you will have 16 bins for each channel and 4096 bins for your 3D color histogram.

- Pick 4 different quantization intervals and give your top-1 accuracy results for each of them on every query dataset.

- Explain the differences in results and possible causes of them if there are any.

Results					
Interval	Bins	Query-1	Query-2	Query-3	Support
16	16	1.0	1.0	0.11	1.0
32	8	1.0	1.0	0.11	1.0
64	4	1.0	1.0	0.12	1.0
128	2	0.935	1.0	0.085	1.0

Query-1 dataset includes scaled versions of images in Support dataset, ie. birds are larger in the pictures. Since scaling is not very extreme, histogram was sucessful to capture the essential pixel distribution for matching the pictures. Query-1 is perfectly detected for all interval sizes except for (128x2) which has larger interval size corresponding to lower spatiality 6 in terms of pixel color values. However, 0.935 of accuracy can still be considered as successfull in Query-1. For intv=128, there is actually a problem and it would be even much severe for the case of intv>128 or intv=256 for the extreme case. It's because of the fact that, scaling up emphasizes bird pixels more, namely there are more bird pixels rather than background pixels as in the Support set. Increasing the interval size decreases the number of bins that pixels can be matched with, hence increasing the difference between support and query-1 as a result of the different pixel density between bird and background.

Query-2 dataset also includes the same images, but with rotations (90, 180, 270 degrees). Since we are calculating histogram for the full image without any regional spatial partitions/grids 6 and it only counts for the frequencies of pixel values, histogram is invariant to rotation in this case. Therefore, as expected Query-2 set perfectly matches to Support set. Note that, in query-2 increasing interval size doesn't decrease the performance because pixel density between bird and background is the same for both rotated and original sets.

Query-3 dataset is composed of same images as in Support dataset without rotation or scaling, but different transformations are applied on these images altering the pixel values. For example, American Pipit turns into yellow from brown. Those transformations might be contrast and hue transformations which can alter the pixel values. As it's clear from the Query-3 set, applying an entire histogram over images wouldn't be sufficient to capture its correspondance in Support set. Altough semantically the images correspond to the same entity, modified pixel values deviate/fool our full histogram approach resulting in poor accuracies for all interval sizes.

Additionally, I've included Support to Support matching for sanity check and comparison as a baseline. As it's clear, it perfectly matches with itself.

2 Per Channel Color histogram

In this section, give your results without dividing the images into grids.

- Pick 5 different quantization intervals and give your top-1 accuracy results for each of them on every query dataset.

- Explain the differences in results and possible causes of them if there are any.

Results					
Interval	Bins	Query-1	Query-2	Query-3	Support
8	32	0.98	1.0	0.125	1.0
16	16	0.98	1.0	0.12	1.0
32	8	0.98	1.0	0.135	1.0
64	4	0.935	1.0	0.14	1.0
128	2	0.585	0.995	0.04	1.0

Overall accuracies are worse than 3D histogram. This is expected because 3D histogram can represent more color combinations, preserving color value spatiality 6. For example, for intv=16 3D histogram has 16x16x16=4096 bins for (R,G,B) to associate pixels whereas per-channel histogram has 16 bins for each channel (R/G/B). In configurations with decreased bin size we expect less accuracy because pixels will be associated/mapped with less choices causing in a sense information loss. The extreme case would be only 1 bin per channel which can be easily fooled, because every histogram will be the same (each pixel fits into 0-255 interval for each channel) regardless of the input that generated that histogram.

Overall accuracies decrease for all queries as interval size increases as a result decreasing spatial 6 or local color representation power.

Query-1 accuracy decreases dramatically for intv=128 with similar reasoning as in 3D histograms background and bird pixel density. However, it decreases much severely compared to 3D histogram which is due to color histogram's weakness for less color spatiality.

Query-2 results are all almost correct except for the intv=128 which was 1.0 in 3D histogram. Once again, per-channel histograms accuracy starts to decrease earlier than 3D histogram's. If we had 256x1 configuration, accuracy loss would be much more dramatic. Query-2 accuracies are higher than Query-1's because of the fact that Query-2 includes the same pixels as in Support set, whereas Query-1 has more

bird pixels which doesn't help the accuracy because our base is Support set with which we measure its KL divergence or distribution similarity.

Query-3 results are still very low, mainly due to the same reasoning as in 3D histogram for this case. In short, pixel values are significantly different. For example, some images are darkened and some brightened: contrast, and hue transformations. These transformations assign pixels different bins compared to support set's. Compared to 3D histogram both methods more or less fail around similar orders of magnitude. However, for this case spatial partitions 6 on images can make a difference which I suspect will increase our representation power in the sense that parts of different images will have more unlikely similarities when cumulated in a histogram.

Before starting the next section, please pick up the best configuration for two properties above and continue with them.

Best Interval Configurations:

		Configuration			
		Query-1	Query-2	Query-3	Support
3D		16	16	64	16
Per-Ch		8	8	64	8

3 Grid Based Feature Extraction - Query set 1

Give your top-1 accuracy for all of the configurations below.

3.1 48×48 spatial grid

- 3d color histogram: 1.0

- per-channel histogram: 1.0

3.2 24×24 spatial grid

- 3d color histogram: 1.0

- per-channel histogram: 1.0

3.3 16×16 spatial grid

- 3d color histogram: 1.0

- per-channel histogram: 0.995

3.4 12×12 spatial grid

- 3d color histogram: 1.0

- per-channel histogram: 0.995

3.5 Questions

- What do you think about the cause of the difference between the results?

- Explain the advantages/disadvantages of using grids in both types of histograms if there are any.

Firstly, almost every grid configuration in query-1 results in perfect accuracy score, almost 1 for each case and also the accuracy score is equal to the score without any grids in part-1. Therefore, in this experiment, we cannot observe which grid partitioning is better for 3d color histograms. However, for per-channel histograms we can observe that the accuracy is slightly better with larger spatial grids, but it's not that significant due to a small change in 3rd decimal. However when it's compared to without partitioning, we can note that it has increased from $0.98 - > 1.0$ which I believe is an improvement for per-channel histograms. The reason why grids improve accuracy in per-channel histograms is because of the spatiality. In other words, histograms corresponding to each partitions of the images will be more unlikely to be matched with wrong images. Because it doesn't decrease its accuracy for the correct image, but for the wrong images smaller spatial partitions are more unlikely to be similar than cumulated or larger partitions ie. 96x96 (whole histogram). These grids are actually the subsets of the whole histogram and if cumulated we can obtain the original result for 96x96. However, the situation would be totally different if there were transformations different than only scaling ie. rotation etc.

As a weakness, we can observe that as grid size gets smaller per-channel accuracy decreases a bit and I believe that would be the general trend for both 3d and per-channel. This is because as we decrease grid size, spatiality increases too much, as an extreme case 1x1 grids correspond to 1 pixel partitions which corresponds to histograms with less points or it can also be interpreted as 1 to 1 pixel checking. It wouldn't be a vigorous method, because query-1 contains scaled images which wouldn't perfectly match with Support set. Therefore, I would consider grid size as a hyperparameter which should be tuned between [1,96]. Once again 3D histograms are more vigorous even in the case of grid-based approach due to its stronger representation power. However, its accuracy should also decrease as grid size gets smaller and smaller.

4 Grid Based Feature Extraction - Query set 2

Give your top-1 accuracy for all of the configurations below.

4.1 48×48 spatial grid

- 3d color histogram: 0.71

- per-channel histogram: 0.355

4.2 24×24 spatial grid

- 3d color histogram: 0.565

- per-channel histogram: 0.16

4.3 16×16 spatial grid

- 3d color histogram: 0.545

- per-channel histogram: 0.1

4.4 12×12 spatial grid

- 3d color histogram: 0.555

- per-channel histogram: 0.1

4.5 Questions

- What do you think about the cause of the difference between the results?

- Explain the advantages/disadvantages of using grids in both types of histograms if there are any.

Firstly, overall I can observe that in query-2 partitioning has always decreased the accuracy for both 3d and per channel histograms. This is an expected behavior, because in query-2 the images are rotated and when these images are partitioned their correspondance in support set would be totally different. For example, a crook has been turned upside down hence its foot pixels are matched with its head pixels which wouldn't construct a vigorous method for prediction and we observe this failure in our poor results.

As it's clear from the reasoning above, partitioning is disadvantageous with 3D and per-channel histograms for query-2 or more generally for images including rotation based transformations.

Additionally, we can observe that as spatial grid size gets smaller accuracy decreases for both 3D and per-channel histograms and it would go much lower if spatial grid size approaches 1x1, this is because of the fact that when we apply rotation and decrease the grid size the relevant parts of the images are less likely to overlap causing totally different histogram distributions.

Another thing to notice that per-channel histogram is more fragile and has lost accuracy much more dramatically compared to 3d histogram. This difference is once again because of the more representation power of 3D histogram over per-channel histogram. If we consider this situation in the context of Query-2, we can observe that 3D color histograms perform better because the similarity between the rotated image and the original image is represented in a more detailed manner, namely if some pixel parts in the partitioned grids match, this makes prediction easier compared to per-channel histogram which wouldn't instead pay that much attention for color spatiality in other words specific color combinations.

5 Grid Based Feature Extraction - Query set 3

Give your top-1 accuracy for all of the configurations below.

5.1 48×48 spatial grid

- 3d color histogram: 0.155

- per-channel histogram: 0.235

5.2 24×24 spatial grid

- 3d color histogram: 0.22

- per-channel histogram: 0.315

5.3 16×16 spatial grid

- 3d color histogram: 0.255

- per-channel histogram: 0.315

5.4 12×12 spatial grid

- 3d color histogram: 0.3

- per-channel histogram: 0.305

5.5 Questions

- What do you think about the cause of the difference between the results?

- Explain the advantages/disadvantages of using grids in both types of histograms if there are any.

Overall we can observe that for both 3D and per-channel histogram methods Query-3 accuracies are around 3 times better compared to previous approaches without any spatial partitions. Firstly, this behavior is no surprise, because it comes from the characteristics of query-3 which basically has some contrast/hue transformation or noise applied on top of the original set. Hence, full histogram approaches are vulnerable, because histograms from the modified pixel values doesn't vigorously match to histograms of the original set. Hence, in query-3 partitioning is advantageous for both 3D and per-channel histograms which improves matching the relevant parts of the images in other words intensifiyng the similarity in lower levels of image partitions as a result denoising image predictions up to a certain extent. For example, noisy image partition containing beak of the bird will be more likely to be matched with the image partition containing the original beak of the bird. Although the partition is noisy, it would be more likely (compared to previous approaches) to be matched with the relevant partitions probably partitions with beaks which is expected to be more similar compared to let's say feet of the bird if the noise is approximately or assumed to be uniform across the entire image.

However, different than our previous experiments in grid based methods, we can observe that for both 3D and per-channel histograms accuracies has increased generally (except for 12x12 for per channel histogram) as spatial grid size decreases. This seems like the general trend, because as we decrease grid size or ie. increase positional spatiality, different images will be even more unlikely to be matched with, because apart from the noise applied on top of the image, the underlying pixels would likely to be matched with the original one's. Noise applied on a smaller region would have less severe effect on deviating our histogram's predictions.

Additionally, different than our previous observations, in this case we observe that per-channel histogram performs better than 3D histogram and it cannot be a lucky coincidence. This phenomena is because of per-channel histograms' less representation power which in a way smoothens histogram values when data is noisy, however 3D histograms try to represent it in a more detailed and sparse manner which is more vulnerable to noise or pixel value alterations. For example, pixel values in 3D histograms will be gathered around a far different space region under considerable amount of noise compared to its region at the original pixel value space which results in less distribution similarity. In query-3 or for noisy inputs cumulative or smoothened approaches like per-channel histogram is more vigorous than detailed representations like 3D histograms.

6 Additional Comments and References

Just to make the analysis in the report more clear. I would like to briefly explain some of the terminologies that I've used to remove ambiguity if there are any.

I've used "spatiality" in different contexts.

Regional/Positional Spatiality : pixels associated with respect to the distance. For example, using grids in histograms to intensify local pixel characteristics.

Color Value Spatiality : pixels associated with respect to color value. For example, increasing number of bins in either 3D or per-channel histogram would increase the spatiality in terms of color values.

Additional Grid Based Feature Extraction for different intervals besides the best pick.

3D - Histogram

Results 3D Grid histogram (intv=16)				
Grid	Query-1	Query-2	Query-3	Support
96 x 96	1.0	1.0	0.11	1.0
48 x 48	1.0	0.71	0.135	1.0
24 x 24	1.0	0.565	0.16	1.0
16 x 16	1.0	0.545	0.215	1.0
12 x 12	1.0	0.555	0.255	1.0
Results 3D Grid histogram (intv=32)				
Grid	Query-1	Query-2	Query-3	Support
96 x 96	1.0	1.0	0.11	1.0
48 x 48	1.0	0.655	0.15	1.0
24 x 24	1.0	0.425	0.155	1.0
16 x 16	1.0	0.4	0.22	1.0
12 x 12	1.0	0.39	0.245	1.0
Results 3D Grid histogram (intv=64)				
Grid	Query-1	Query-2	Query-3	Support
96 x 96	1.0	1.0	0.12	1.0
48 x 48	1.0	0.555	0.155	1.0
24 x 24	1.0	0.36	0.22	1.0
16 x 16	1.0	0.3	0.255	1.0
12 x 12	1.0	0.31	0.3	1.0
Results 3D Grid histogram (intv=128)				
Grid	Query-1	Query-2	Query-3	Support
96 x 96	0.935	1.0	0.085	1.0
48 x 48	0.995	0.28	0.15	1.0
24 x 24	1.0	0.195	0.24	1.0
16 x 16	0.995	0.14	0.285	1.0
12 x 12	1.0	0.145	0.34	1.0

Per-Channel Histogram

Results Per-Channel Grid histogram (intv=8)				
Grid	Query-1	Query-2	Query-3	Support
96 x 96	0.98	1.0	0.125	1.0
48 x 48	1.0	0.355	0.215	1.0
24 x 24	1.0	0.16	0.22	1.0
16 x 16	0.995	0.1	0.24	1.0
12 x 12	0.995	0.1	0.24	1.0

Results Per-Channel Grid histogram (intv=16)				
Grid	Query-1	Query-2	Query-3	Support
96 x 96	0.98	1.0	0.12	1.0
48 x 48	1.0	0.365	0.21	1.0
24 x 24	1.0	0.17	0.225	1.0
16 x 16	1.0	0.115	0.25	1.0
12 x 12	0.995	0.11	0.245	1.0

Results Per-Channel Grid histogram (intv=32)				
Grid	Query-1	Query-2	Query-3	Support
96 x 96	0.98	1.0	0.135	1.0
48 x 48	1.0	0.36	0.22	1.0
24 x 24	1.0	0.205	0.26	1.0
16 x 16	1.0	0.13	0.27	1.0
12 x 12	0.995	0.1	0.265	1.0

Results Per-Channel Grid histogram (intv=64)				
Grid	Query-1	Query-2	Query-3	Support
96 x 96	0.935	1.0	0.14	1.0
48 x 48	1.0	0.33	0.235	1.0
24 x 24	1.0	0.2	0.315	1.0
16 x 16	0.995	0.125	0.315	1.0
12 x 12	0.995	0.1	0.305	1.0

Results Per-Channel Grid histogram (intv=128)				
Grid	Query-1	Query-2	Query-3	Support
96 x 96	0.585	0.995	0.04	1.0
48 x 48	0.97	0.175	0.21	1.0
24 x 24	1.0	0.125	0.355	1.0
16 x 16	0.985	0.1	0.395	1.0
12 x 12	0.98	0.065	0.405	1.0