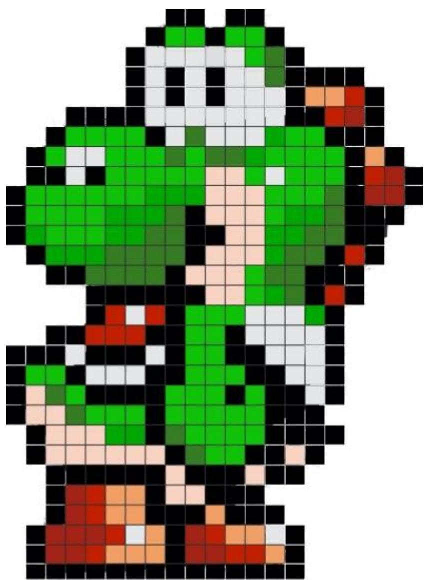
INFS4205/7205 Advanced Techniques for High Dimensional Data

Tutorial 9: An Introduction to Multimedia Databases

Semester 1, 2020

Question 1: Suppose we are using 24-bit RGB model for image representation, and extract color features using the 4-bit 3D color histogram: 2 bits for red, and 1 bit for green and blue. Given a picture of the Yoshi (21 × 29 bit), what is the color distribution of it?



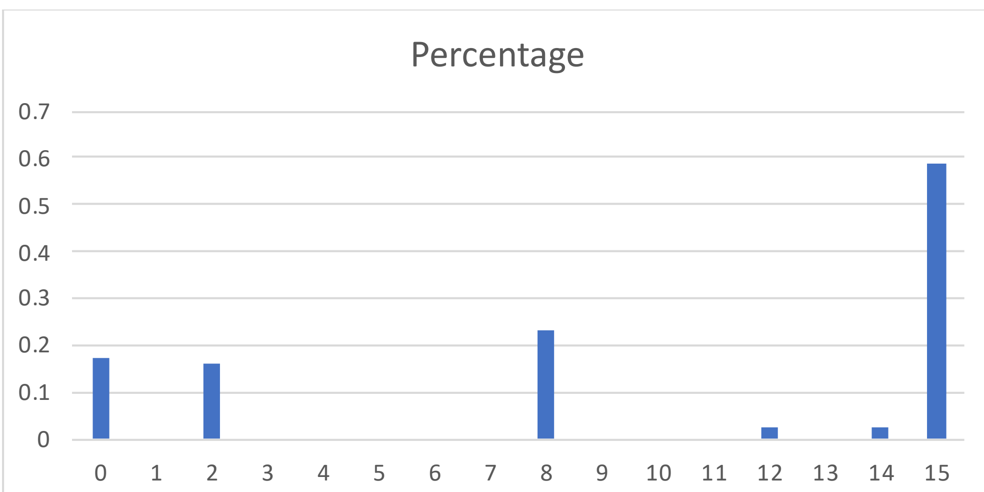
Sample Solution:
Firstly, this figure has following pixels (suppose I counted correctly…)

1. (22, 191, 9), 78 pixels, (0,1,0)
2. (0, 169, 0), 22 pixels (0,1,0)
3. (57, 125, 39), 30 pixels (0,0,0)
4. (246, 211, 193), 35 pixels (3,1,1)
5. (239, 160, 101), 15 pixels (3,1,0)
6. (193, 32, 1), 17 pixels (3,0,0)
7. (163, 35, 23), 14 pixels (2,0,0)
8. (0, 0, 0), 75 pixels (0,0,0)
9. (229, 230, 231), 46 pixels (3,1,1)
10. (255,255,255), 277 pixels (3,1,1)

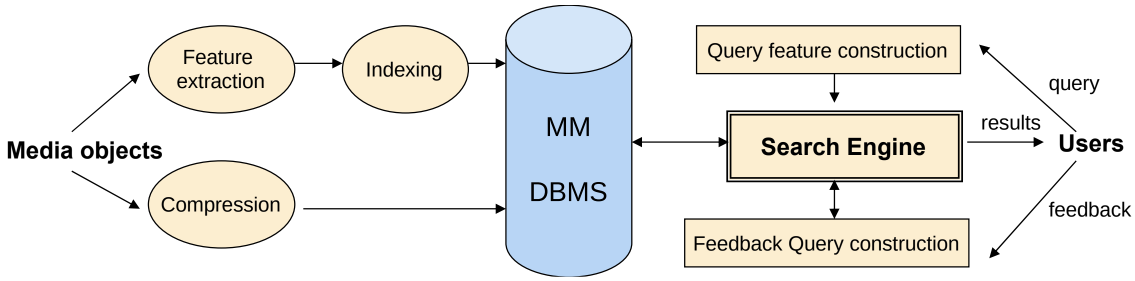
Next we fill the 3D histogram table to do the counting:

Bin	R,G,B	# Pixels	Percentage
0	0,0,0	105	0.172
1	0,0,1	0	0
2	0,1,0	100	0.164
3	0,1,1	0	0
4	1,0,0	0	0
5	1,0,1	0	0
6	1,1,0	0	0
7	1,1,1	0	0
8	2,0,0	14	0.23
9	2,0,1	0	0
10	2,1,0	0	0
11	2,1,1	0	0
12	3,0,0	17	0.028
13	3,0,1	0	0
14	3,1,0	15	0.025
15	3,1,1	358	0.588

Finally, we can get the histogram below:



Question 2: Discuss the architecture of the multimedia database system.



Example Solution:
On the left-hand side is the storage / index component, and on the right-hand side is the query component.

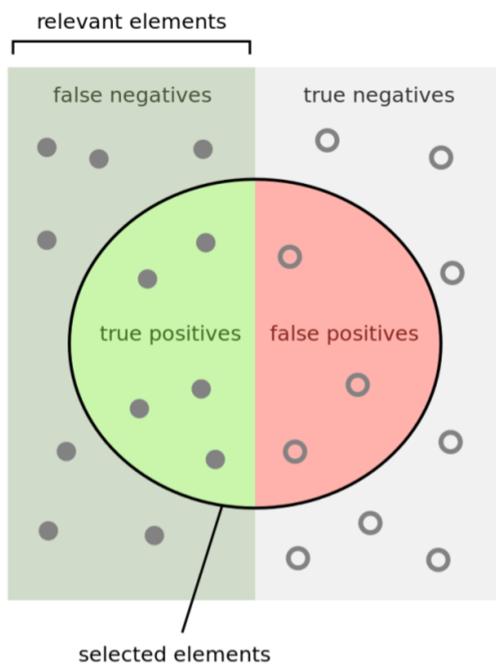
For each multimedia object to be stored in the MMDBMS, we stores its compressed version in disk because most of the MM data are very big. This compressed data is only used to return as the final result, and it will not be used during the search. Therefore, in order to support query/search, we need to extract features from the MM object and organize these features. This process is called feature extraction. For most of the cases, the extracted features have a large dimensional number, so we treat them as high dimensional data and organize them with the high dimensional indexes. Once again, the index is built on the features, not on the original data. Each original MM object is represented by a point in the feature space. How the features are abstracted is also part of the MMSDMBS design, so all the data stored in it share the same feature space.

During the search, user provides a query object. MMDBMS does not use this query object to search directly. Instead, it first abstracts the features of the query object with exactly the same process when it abstracts the data it stores. Then it uses the query features to search in the feature space with the help of high dimensional index. With the algorithm of kNN, it finds a list of points that are similar to the user's query object, and retrieves the actual data stored in the database, decompresses them, and returns them to the user.

There is feedback loop for user to evaluate the query result and let the database refine the result. Because all the results are feature similarity-based, there is no guarantee the results can satisfy the user's need. Therefore, this feedback loop can help identify which feature the user cares more and assigns higher weights to them during the search. For example, if the user cares more about the color, then in the next iteration, the search gives higher weight on the color feature and lower weights on the other features like shape and texture.

Question 3: Suppose we have 100 data in the database, 60 of them are positive, 40 of them are negative, and we what to retrieve all the positive ones. However, the system returns 50 results, and 40 of them are positive. What are the precision and recall of this result?

Sample Solution:



As shown in the above plot, the overall rectangle contains the entire data set, with the dots (left half) representing the positive data, and circles (right half) representing the negative data. The big circle is the result of query finding all the positive data. The dots in the green half are the correct ones, so they are call *true positive* (*true* because of correct, *positive* because the query asks for positive). In our example, the true positive number is 40.

The circles in the red half appear in the positive results, but they are actually negative, so they are not correct. Therefore, we call them *false positive* (*false* because they are incorrect, *positive* because the query asks for positive). In our example, the false positive number is 10.

For all the data outside the result circle, the query result regards them as negative. However, on the left-hand side dots, they are positive, so this negative is not correct, and we call them *false negative*. In our example, it is 20.

Similarly, the right circles are actually negative, so we call them *tree negative*. In our example, it is 30.

The precision asks for the ratio of the correctly returned ones in the returned results, so it is $Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{40}{50} = 0.8$.

The recall asks for the ratio of the correctly returned ones in all the correct ones, so it is $Recall = \frac{True\ Positive}{True\ Positive + False\ negative} = \frac{40}{60} = 0.667$.

Question 4: We always want to achieve high precision and high recall in a system, but it is not always achievable. Please discuss the relation between the precision and recall.

Sample Solution:
Often, there is an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other. Brain surgery provides an illustrative example of the trade-off. Consider a brain surgeon tasked with removing a cancerous tumour from a patient's brain. The surgeon needs to remove all of the tumour cells since any remaining cancer cells will regenerate the tumour. Conversely, the surgeon must not remove healthy brain cells since that would leave the patient with impaired brain function. The surgeon may be more liberal in the area of the brain she removes to ensure she has extracted all the cancer cells. This decision increases recall but reduces precision. On the other hand, the surgeon may be more conservative in the brain she removes to ensure she extracts only cancer cells. This decision increases precision but reduces recall. That is to say, greater recall increases the chances of removing healthy cells (negative outcome) and increases the chances of removing all cancer cells (positive outcome). Greater precision decreases the chances of removing healthy cells (positive outcome) but also decreases the chances of removing all cancer cells (negative outcome). Therefore, another performance measure, *F-measure*, is proposed to combine precision and recall by their harmonic mean, as shown below:

$$F = 2 \times \frac{precision \times recall}{precision + recall}$$