University Kasdi Merbah-Ouargla
Departement of computer science and TI
Module : TIVO
Duration : 1 H
17/05/2022

EXAM

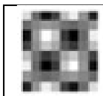
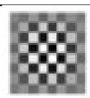
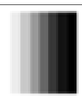
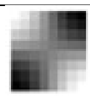
R1. **(2 PTS, 0.25 for each edge pixel)** After convolving an image using Laplacian of Gaussian (LoG) we get the image below, locate the edge pixels

12.3	3.8	7.9	-51.3
15.6	1.4	-5.3	59.2
19.8	-1.5	6.9	56.7
-2.3	2.3	-8.1	-49.6

R2. **(2 PTS, 0.25 for each pair of edge pixels)** Given the following magnitude image, if you know that $TLow = 12$ and $THigh = 71$ use the hysteresis thresholding to locate the edge pixels (canny)

12.3	3.8	7.9	51.3	87	95.2	40.5
9	97	15	75.6	52	91.3	17
5	3	99.2	25	69.3	33.5	12.5
4.2	100	8	100.4	65.4	89	35
15.6	1.4	5.3	59.2	16	88	17.5
19.8	1.5	6.9	56.7	10	29	70
2.3	2.3	8.1	49.6	9.3	15	78

R3. **(2 PTS, 0.5 for each coefficient)** Given the following DCT filters and given the 4 coefficients A, B, C and D, such that $A > B > C > D$, match each filter to the suitable coefficient?

			
Filter 1 **C**	Filter 2 **D**	Filter 3 **A**	Filter 4 **B**

R4. **(1.5 PTS, 0.25 for each partition + 0.5 for math concept)** Given a set of three data-points $s = \{1,2,3\}$, enumerate all the possible clustering solutions for this set? If we add s to the set of solutions, what we call this set in mathematics?

L'ensemble {1, 2, 3} a les partitions suivantes :

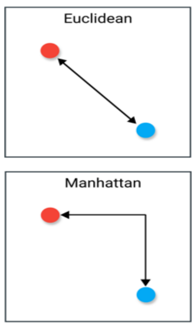
- { {1}, {2}, {3} } ;
- { {1, 2}, {3} } ;
- { {1, 3}, {2} } ;
- { {1}, {2, 3} } ;
- ~~et { {1, 2, 3} }.~~

Mathematically is called partition of a set.

R5. (2 PTS, 0.5 for the example + 1.5 for the geometric interpretation) Suppose we are given the following 2D data-points $\{(N, 1), (M, 1), (T, 1), (B, 1), (V, 1)\}$, we want to pick up the suitable distance for k-means-based clustering. Demonstrate by a simple example that Euclidean and Manhattan distances are equivalent in this case? Explain this equivalence by a by a geometrical illustration? Euclidean and Manhattan distances are given by

$$D_M(x, y) = \sum_{i=1}^{N} |x_i - y_i| \qquad\qquad D_E(x, y) = \sqrt{\sum_{i=1}^{N} |x_i - y_i|^2}$$

E.g., Dm((N,1),(M,1) = |N-M|, De((N,1),(M,1) = |N-M|,



In this case there will be no diagonal difference between the two data-points because they lie on the same line, thus the two distances are equivalent.

R6. (3 PTS, 1.5 for each matrix) Given the following images

3	8	7
5	4	3
9	1	6

1	4
8	5

Scale up the 3x3 image to 7x7 using k-nearest neighbor interpolation, and scale up the 2x2 to 5x5 using bilinear interpolation? (include the calculations for the bilinear interpolation)

3	3	8	**8**	8	7	**7**
3	3	8	8	8	7	7
5	5	4	4	4	3	3
5	5	4	**4**	4	3	**3**
5	5	4	4	4	3	3
9	9	1	1	1	6	6
9	9	1	**1**	1	6	**6**

1	X1	X2	**4**	NAN
X3	X4	X5	X6	NAN
X7	X8	X9	X10	NAN
8	X11	X12	**5**	NAN
NAN	NAN	NAN	NAN	NAN

X1 = 1*2/3 + 4*1/3 = 1.99 X2 = 1*1/3 + 4*2/3 = 2.99

X3 = 1*2/3 + 8*1/3 = 3.32 X7 = 1*1/3 + 8*2/3 = 5.66

X6 = 4*2/3 + 5*1/3 = 4.33 X10 = 5*2/3 + 4*1/3 = 4.66

X11 = 8*2/3 + 5*1/3 = 7 X12 = 5*2/3 + 8*1/3 = 6

X4 = X1*2/3 + X11*1/3 = 1.99*2/3 + 7*1/3 = 1.32 + 2.33 = 3.65

X5 = X2*2/3 + X12*1/3 = 2.99*2/3 + 6*1/3 = 1.99 + 2 = 3.99

X8 = X11*2/3 + X1*1/3 = 7*2/3 + 1.99*1/3 = 4.66 + 0.66 = 5.32

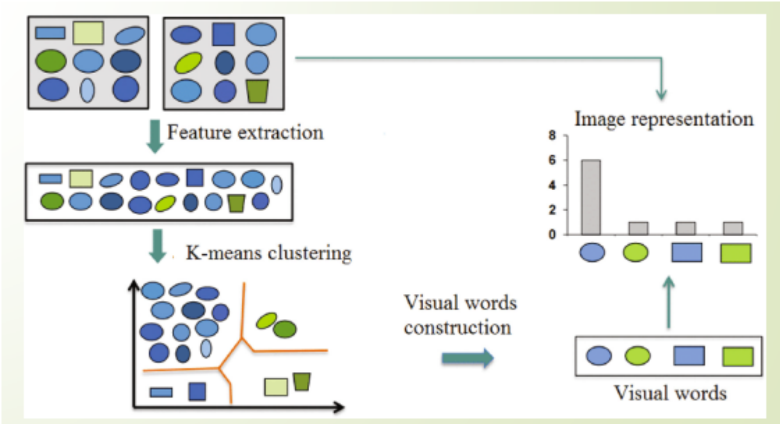
X9 = X12*2/3 + X2*1/3 = 6*2/3 + 2.99*1/3 = 4 + 0.99 = 4.99

1	2	3	**4**	NAN
3	4	4	4	NAN
6	5	5	5	NAN
8	7	6	**5**	NAN
NAN	NAN	NAN	NAN	NAN

R7. (2 PTS) In HOG computation, the gradient magnitude and orientation for a specific pixel are found to be **21** and **45**, respectively. The amount of magnitude is divided between two neighboring bins, identify those bins and the amount to put in each of them?

Mag			(60-45)/20*21 = 15.75	(45-40)/20*21 = 5.25					
Bin	0	20	40	60	80	100	120	140	160

R8. (1.5 PTS) The number of key-points (detected using SIFT) in two different images may be different (i.e., dimension of the two feature vectors is different). Explain briefly how can we use Bag of Words (BOW) to match the two images in this case?



Key-points from all the images are gathered together and then clustered using k-means. Visual words (= number of clusters) are the centroids of the resulting clusters. Each key-point is assigned to one cluster, and a histogram is generated for each image (dimension = number of clusters), where each bin counts the image key-points that belong to that bin. After that, the new representation can be fed to the conventional classifiers e.g., SVM, KNN…etc.

R9. **(2 PTS)** Given the following confusion matrix, which reports the performance of SVM classifier in classifying persons to sick (positive) and healthy (negative).

4	2
3	5

- Translate the matrix into text?
- Calculate the accuracy, specificity and sensitivity?
- Determine the number of persons that are classified?

TP = 4 persons are sick and identified by the classifier to be sick

TN = 5 persons are healthy and identified by the classifier to be healthy

FP = 2 persons are healthy and identified by the classifier to be sick

FN = 3 persons are sick and identified by the classifier to be healthy

Accuracy = (TP + TN) / (TP + TN + FP + FN) = 9/14 = 0.64

Specificity = TN / (FP + TN) = 5 / (5+2) =5/7 = 0.71

Sensitivity = TP / (FN + TP) = 4 / (4+3) = 4/7 = 0.57

Number of persons = 4 + 2 + 3 + 5 = 14

R10. **(2 PTS)** Prove that correlation coefficient σ_{xy} of two image vectors in SSIM is equivalent to the cosine similarity between the two vectors? Cosine similarity of two N-dimensional vectors and σ_{xy} are respectively given by

$$cos(\theta) = \frac{\sum_{i=1}^{N} y_i \,.\, x_i}{\sqrt[2]{\sum_{i=1}^{N} y_i^2}\,\sqrt[2]{\sum_{i=1}^{N} x_i^2}} \qquad \sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \mu_y)(x_i - \mu_x)$$

N.B. σ_{xy} is calculated after subtracting the mean and normalizing the variance i.e., the vector images will be $\frac{(x-\mu_x)}{\sigma_x}$ and $\frac{(y-\mu_y)}{\sigma_y}$, respectively.

$y' = \frac{(y-\mu_y)}{\sigma_y}$ $x' = \frac{(x-\mu_x)}{\sigma_x}$, for both x' and y', $\mu_{x'} = 0$, $\sigma_{x'} = 1$, and $\mu_{y'} = 0$, $\sigma_{y'} = 1$

$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^{N} (y'_i - \mu_y)(x'_i - \mu_x)$ and $\cos(\theta) = \frac{\sum_{i=1}^{N} y'_i \,.\, x'_i}{\sqrt[2]{\sum_{i=1}^{N} y'^2_i}\,\sqrt[2]{\sum_{i=1}^{N} x'^2_i}}$

In other words, we have to prove that

$\sum_{i=1}^{N} (y'_i - \mu_y)(x'_i - \mu_y) = \sum_{i=1}^{N} y'_i \,.\, x'_i$ **(1)** and

$\sqrt[2]{\sum_{i=1}^{N} y'^2_i}\,\sqrt[2]{\sum_{i=1}^{N} x'^2_i} = N - 1$ **(2)**

(1) We have $\mu_{x'} = \mu_{y'} = 0$, then, simply

$\sum_{i=1}^{N} (y'_i - \mu_{y'})(x'_i - \mu_{x'}) = \sum_{i=1}^{N} y'_i \,.\, x'_i$

(2) Now, we have the variance formula

$\frac{\sum_{i=1}^{N} (y'_i - \mu_y)^2}{N-1} = 1$ as $\sigma_{x'} = 1$ and $\sigma_{y'} = 1$, thus,

$\sum_{i=1}^{N} (y'_i - \mu_{y'}) = N - 1$, thus, $\sum_{i=1}^{N} (y'_i) = N - 1$, because $\mu_{y'} = 0$

$\sqrt[2]{\sum_{i=1}^{N} (y'_i)} = \sqrt[2]{N-1}$ from this we can deduce that

$\sqrt[2]{\sum_{i=1}^{N} (y'_i)} \,.\, \sqrt[2]{\sum_{i=1}^{N} (x'_i)} = N - 1$