Submission Deadline: @17:59 9 December 2021

| No. | Points |
|---|---|
| 1 | /14 |
| 2 | /24 |
| 3 | /12 |
| TOTAL | /50 |

# Answer sheet for 1/2564 Final Exam for
# 2110431 Introduction to Digital Imaging
# 2147329 Digital Image Processing and Vision Systems

**Read the instructions carefully**

**Rules of Conduct for Students during Examinations**

Unacceptable examination conduct includes

1) Communicating with other students or anyone regarding the exam.
2) Looking at exam materials of other students or anyone or allowing others to look at their exam materials.
3) Copy some or all of the answer from internet searches. (Suggest to write in your own word)

When a student cheats or is in suspicion, the supervisory officer has the power to investigate the matter. There are serious consequences to academic misconduct, including receiving an “F” on the work or examination, an “F” in the course.

Name_______Panithi Vanasirikul______________ID___6230309321___

You must submit

**(1) Answer sheet (in pdf)**

**(2) .py code for problem 2**

In MyCourseVille before 17:59, December 9, 2021. Otherwise, the exam will not be accepted.

THE EXAM IS **DUE** ON 17:59, DECEMBER 9, 2021

**Problem 1**

Fill in your ID here

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **6** | **2** | **3** | **0** | **3** | **0** | **9** | **3** | **2** | **1** |
| **$ID_0$** | **$ID_1$** | **$ID_2$** | **$ID_3$** | **$ID_4$** | **$ID_5$** | **$ID_6$** | **$ID_7$** | **$ID_8$** | **$ID_9$** |

Fill in your 4x10 quantized image here

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **2** | **1** | **1** | **0** | **1** | **0** | **3** | **1** | **1** | **0** |
| **2** | **1** | **1** | **0** | **1** | **0** | **3** | **1** | **1** | **0** |
| **0** | **1** | **1** | **3** | **0** | **1** | **0** | **1** | **1** | **2** |
| **0** | **1** | **1** | **3** | **0** | **1** | **0** | **1** | **1** | **2** |

1.1.1 Determine total mean ($\mu_T$)

Total mean ($\mu_T$) = 0*(12/40) + 1*(20/40) + 2*(4/40) + 3*(4/40) = 1

1.1.2 Determine the values in the table

| **Intensity Level $i$** | **0** | **1** | **2** | **3** |
|---|---|---|---|---|
| $n_i$ | 12 | 20 | 4 | 4 |
| $p_i$ | 0.3 | 0.5 | 0.1 | 0.1 |

For each selected threshold level $k$, calculate the between-class variance $\sigma_B^2(k)$ and other parameters below.

| **Threshold level** $k$ | **0** | **1** | **2** | **3** |
|---|---|---|---|---|
| $\omega_0(k)$=$\sum_{i=0}^{k} p_i$ | 0.3 | 0.8 | 0.9 | |
| $\mu(k) = \sum_{i=0}^{k} ip_i$ | 0 | 0.5 | 0.7 | |
| $\omega_1(k) = 1 - \omega_0\ (k)$ | 0.7 | 0.2 | 0.1 | |
| Between-Class Variance $\sigma_B^2(k)$ | 0.4286 | 0.5625 | 0.4444 | |

1.1.3 The selected threshold $k$ or Otsu's threshold = 1

1.1.4 Fill in values of the thresholded image after apply inverse thresholding using this Otsu's threshold by **highlighting black color if the value = 0, and remain white box for value = 1 (or 255).**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **0** | **1** | **1** | **1** | **1** | **1** | **0** | **1** | **1** | **1** |
| **0** | **1** | **1** | **1** | **1** | **1** | **0** | **1** | **1** | **1** |
| **1** | **1** | **1** | **0** | **1** | **1** | **1** | **1** | **1** | **0** |
| **1** | **1** | **1** | **0** | **1** | **1** | **1** | **1** | **1** | **0** |

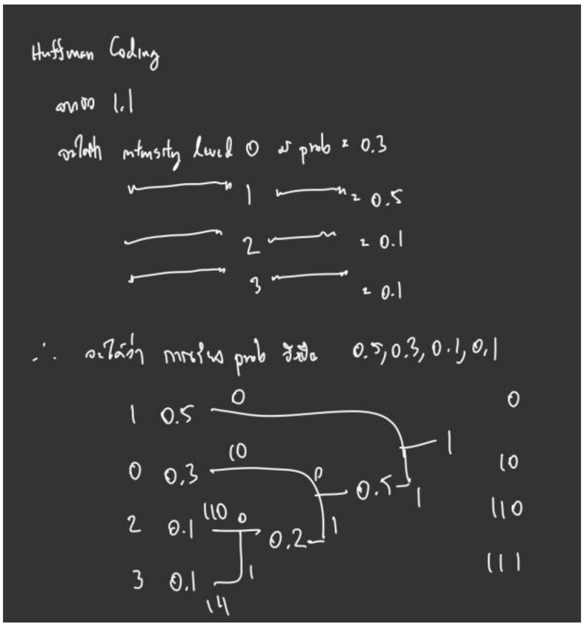
## 1.2 Image Compression

### 1.2.1 Minimum fixed bits (code #1)

| Intensity level ($r_k$) | Code #1 |
|---|---|
| 0 | 00 |
| 1 | 01 |
| 2 | 10 |
| 3 | 11 |

** Select code #1 should be a binary increment corresponding to the intensity levels.

### 1.2.2 Huffman coding (code #2)




For any level has the same probabilities, then also rank them by intensity levels.

| Intensity level ($r_k$) | Code #2 | Probability of intensity level ($r_k$) $P(r_k)$ |
|---|---|---|
| 0 | **10** | **0.3** |

| | | |
|---|---|---|
| 1 | **0** | **0.5** |
| 2 | **110** | **0.1** |
| 3 | **111** | **0.1** |

1.2.3 Encode this image sequence below using Code #1 & Code #2

| $ID_0$ | $ID_1$ | $ID_2$ | $ID_3$ | $ID_4$ | $ID_5$ | $ID_6$ | $ID_7$ | $ID_8$ | $ID_9$ |
|---|---|---|---|---|---|---|---|---|---|

Fill in your image sequence:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **6** | **2** | **3** | **0** | **3** | **0** | **9** | **3** | **2** | **1** |

| $qID_0$ | $qID_1$ | $qID_2$ | $qID_3$ | $qID_4$ | $qID_5$ | $qID_6$ | $qID_7$ | $qID_8$ | $qID_9$ |
|---|---|---|---|---|---|---|---|---|---|

Fill in your quantized image sequence:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **2** | **1** | **1** | **0** | **1** | **0** | **3** | **1** | **1** | **0** |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Code #1** | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | | | | |
| **Code #2** | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| | 1 | 0 | 0 | 1 | 0 | | | | | | | |

**Problem 2**

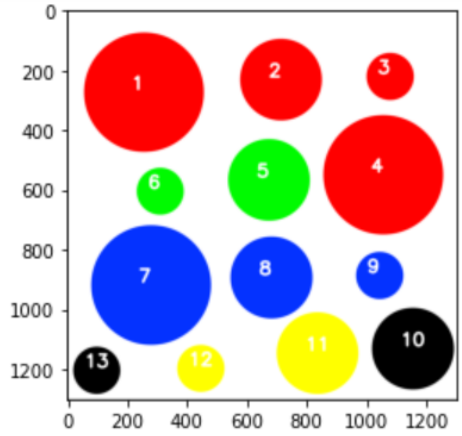
Don't forget to upload .py in MyCourseVille

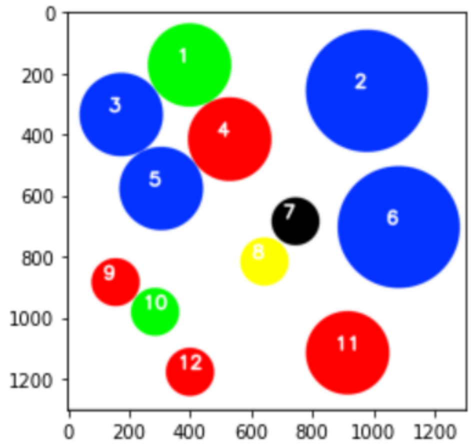
2.1) Describe steps of your implementation and key parameters

| Steps | Description and purposes |
|---|---|
| 1 | Converting the color to gray scale to be used for doing binary threshold |
| 2 | Using inverse binary threshold to cut noises (if any) and creates an output image with only 0 and 1 in array |
| 3 | Using erosion in each of the methods to separate some components that are too close to each other |
| 4 | Use cv2.connectedComponentsWithStats to get number of connected components |
| 5 | In case the method is for counting specific shapes, we loop the connected components and use another method(image_type) which precalculated the number of pixels in each type of size(small, medium, large) and shape(circle, triangle, rectangle, umbrella) to check whether the connected component we are looping has the number of pixels within the range or not.<br>(Basically, checking whether each size and shape has its own unique range of number of pixels or not) |
| 6 | The loop for connected components write texts at the centroid of components indicating the count of each specified method to visualize where the code could detect each shape and size within interest |

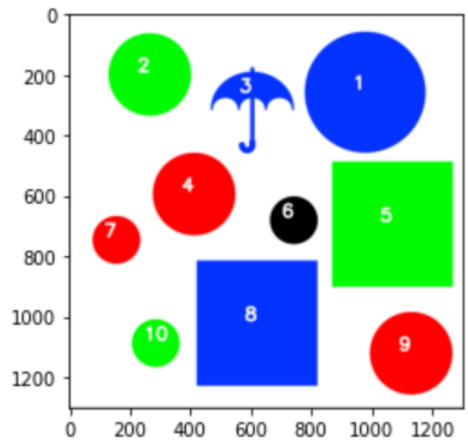
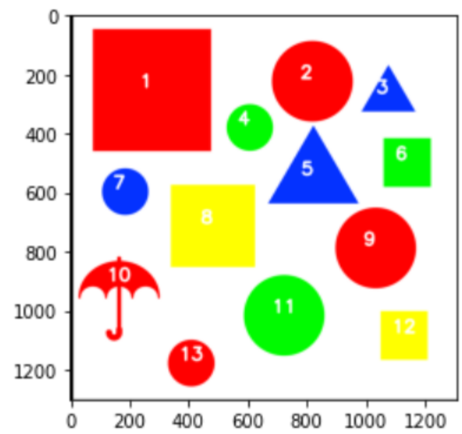
2.2) Capture results

| Image | Results |
|---|---|

<table>
<tr><td></td><td>(Capture the num_count results of your implemented functions here)</td></tr>
<tr><td>0000.png</td><td>



Num of all objects:  13<br>
Num of circles:  13<br>
Num of small circles:  5<br>
Num of medium circles:  5<br>
Num of large circles:  3<br>
Num of rectangles:  0<br>
Num of triangles:  0<br>
Num of umbrella:  0<br>
-----------------------------------
</td></tr>
<tr><td>0001.png</td><td>


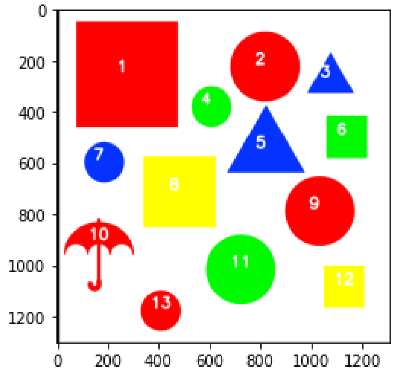

Num of all objects:  12<br>
Num of circles:  12<br>
Num of small circles:  5<br>
Num of medium circles:  5<br>
Num of large circles:  2<br>
Num of rectangles:  0<br>
Num of triangles:  0<br>
Num of umbrella:  0<br>
------------------------------------
</td></tr>
</table>

| | |
|---|---|
| 0002.png | <br>Num of all objects: 10<br>Num of circles: 7<br>Num of small circles: 3<br>Num of medium circles: 3<br>Num of large circles: 1<br>Num of rectangles: 2<br>Num of triangles: 0<br>Num of umbrella: 1<br>------------------------------------ |
| 0003.png | <br>Num of all objects: 13<br>Num of circles: 6<br>Num of small circles: 3<br>Num of medium circles: 3<br>Num of large circles: 0<br>Num of rectangles: 4<br>Num of triangles: 2<br>Num of umbrella: 1<br>------------------------------------ |

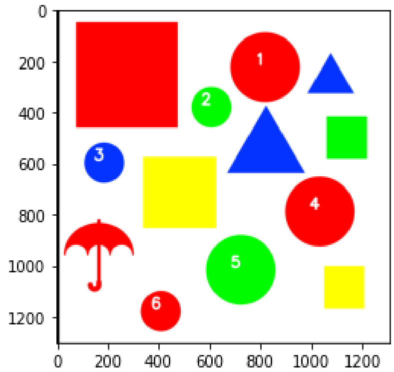
2.3) Explain and show how you represent the result_image for each case. Show examples.

I chose to represent the result_image with a text counting from 1 on each centroid of the shapes within interest. Therefore the numbers on each image for method count_all_objs corresponds to the number of connected components directly and the other methods corresponds to num_count.

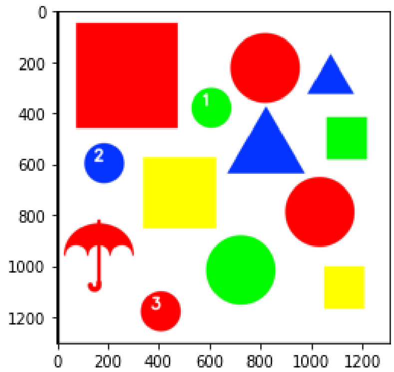
For example, if I use 0003.png, the results of result_image of each method would be as shown below:



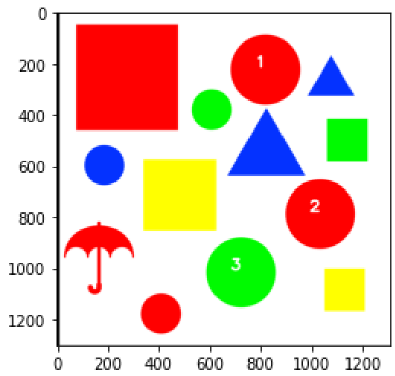

```
Num of all objects:  13
```



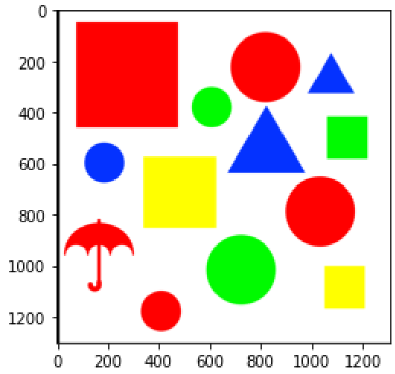

```
Num of circles:  6
```



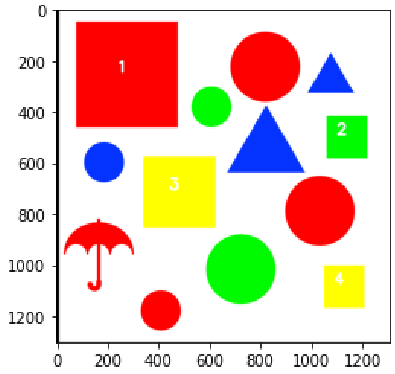

```
Num of small circles:  3
```



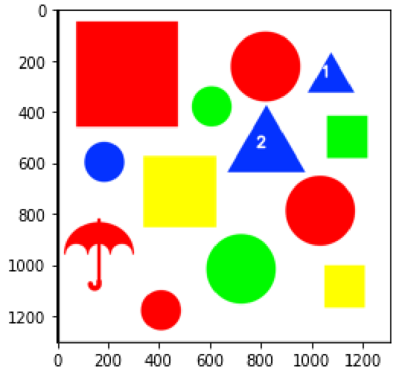

```
Num of medium circles:  3
```



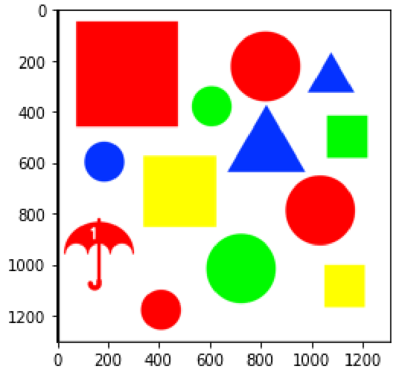

```
Num of large circles:  0
```




```
Num of rectangles:  4
```




```
Num of triangles:  2
```




```
Num of umbrella:  1
----------------------------------------------
```

## Problem 3 (< 350 words)

MRI and PET coordination

The problem for the MRI Scan and PET Scan is that the analysis for the subject must be in the same coordinate space. Bye warp 2 images correctly into the same coordinate space, the 2 images need to have their contrast and their outlines correlated sufficiently. Moreover, PET can be a dynamic data (PET takes around 15-90 minutes) meaning that early, mid, and late time frames are likely to be very different images.

The algorithm for the whole process should be:

1. Creating a new PET image that can represent all the dynamic PET images and the MRI image
2. Mapping the dynamic PET images to our created PET image and map our created PET image to the MRI image

Techniques

Image Alignment

Idea: There are many algorithms within image alignment used depending on the aspects and the scope of our interest such as different light directions, different sensors, different angles etc. but the idea for the algorithm is the same which is to transform images to reference image using a coordinate system. By creating coordinate system between all the PET images, creating an image that describes its coordinate system which existing images can warp into would solve the first part of the algorithm. Image Alignment is widely used in motion analysis and computer vision.

Limitations: Feature detection is the key to aligning multiple images into the same coordinate space. If the features used is not varied or not invariant, some features would be left out hence the warping to the coordinate space is possible to inconsistency.

Image Co-registration

Idea: Image co-registration is classified into intensity-based and feature-based. For intensity-based registrations, the study involves intensity patterns which from my understanding, is the difference between PET and MRI scan. By using intensity-based registrations, we can successfully create a correlation between the 2 scans.

Limitations: There will be an uncertainty level created from registering images having any spatial differences. Therefore, whenever we use the co-registration algorithm, we must consider the uncertainty level and how critical it could be in analyzing the synthesized data onwards.

References:

Misciagna, Sandro (2013). *Positron Emission Tomography - Recent Developments in Instrumentation, Research and Clinical Oncological Practice || Basic PET Data Analysis Techniques.*

Princeton University (2014). *Alignment of Images Captured Under Different Light Directions.*

Yang, X. (2017). *Uncertainty Quantification, Image Synthesis and Deformation Prediction for Image Registration.*