Petra University		جامعة البترا - خمسة وعشرون عاما University of Petra Anniversary
Private Accredited University		(جامعة خاصة معتمدة)
Faculty of Information Technology		كلية تكنولوجيا المعلومات

Computer Vision(606384_1)

Midterm Exam

Department: DSAI

Academic Year: 2025-2026

Term: 1st

Date: December 14, 2025

Instructor(s): Dr. Mohammad Al-Oudat

Coordinator: Dr. Mohammad AL-Oudat

Student's Name: _____

Student's ID: _____

Section: 1

Question	*Points*	*Score*
1	*Out of 10*	
2	*Out of 5*	
3	*Out of 5*	
4	*Out of 10*	
Total	*30*	

Exam Notes and Rules:

- *The following material is provided: exam paper.*
- *Materials allowed: none.*
- *The exam period is 60 minutes.*
- *Be very specific and brief in your answers.*
- *Show all work for full credit.*

QUESTION 1: [10 POINTS]

Choose the best answer:

1. What is an image made of?
    a. Only brightness values
    b. A matrix of pixels
    c. A list of text values
    d. Audio samples

2. What does converting an image to grayscale do?
    a. Adds more color
    b. Makes the image 3-channel
    c. Reduces it to one channel (0–255 values)
    d. Converts it to binary

3. Which filter is BEST for removing salt-and-pepper noise?
    a. Gaussian Blur
    b. Median Blur
    c. Sharpening
    d. Histogram Equalization

4. Which smoothing method gives the most natural blur?
    a. Average Blur
    b. Median Blur
    c. Gaussian Blur
    d. Canny Blur

5. What does histogram equalization improve?
    a. File size
    b. Color saturation
    c. Image contrast
    d. Sharpness

6. Which thresholding technique automatically finds the best cutoff?
    a. Simple Thresholding
    b. Adaptive Thresholding
    c. Otsu's Thresholding
    d. Binary Inversion.
7. Which edge detection method produces the cleanest and most reliable edges?
    a. Sobel
    b. Laplacian
    c. Canny
    d. Histogram

8. Which OpenCV function is used to combine Sobel X and Y results?

a. cv2.merge()
b. cv2.addWeighted()
c. cv2.stack()
d. cv2.add()

9. Why do we resize images in preprocessing?
    a. To change image color
    b. To reduce computation and standardize input size
    c. To add more details
    d. To rotate the image

10. A binary image contains only pixel values 0 or 1
    a. True
    b. False

QUESTION 2: [5 POINTS]

Replace the substituted Parmeter (**?**) with the correct term, and describe the purpose of it:

1. $cv2.rectangle(img, (?,100), (410,410), (0,255,0), 4)$

    Correct Term: 100, x-coordinate (distance from left).

2. $cv2.circle(img, (150,150), ?, (255,0,0), 2)$

    Correct term: integer number, radius of the circle (in pixels)

3. $cv2.putText(img, \text{"Hello"}, (60,40), cv2.FONT\_HERSHEY\_SIMPLEX, 1, ?, 2)$

    (0,0,255) → the text color in BGR format

4. $cv2.convertScaleAbs(img, ?, beta=30)$

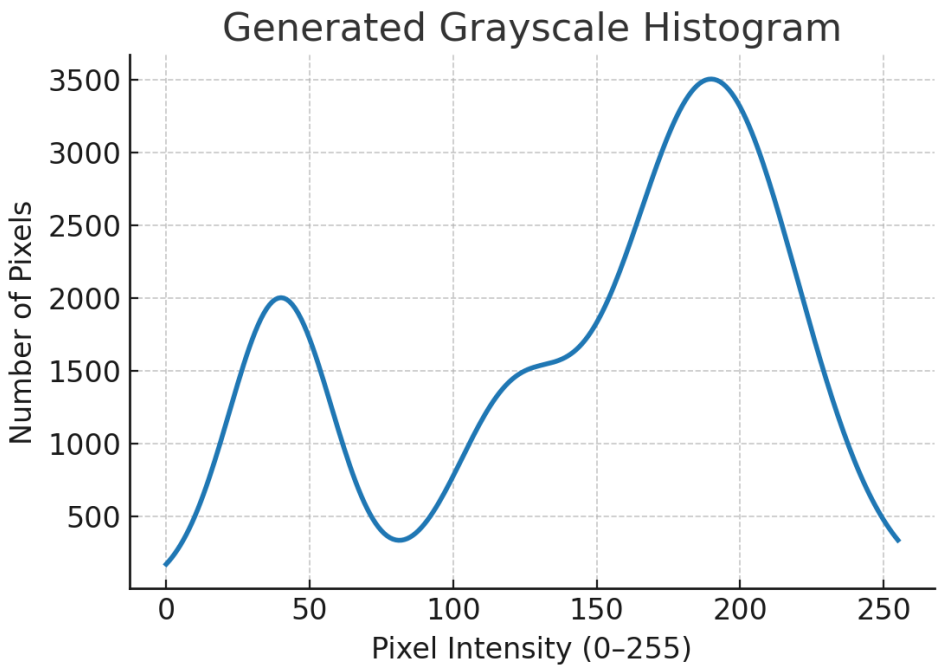
    alpha=1.2, beta=30 → slightly sharper & brighter.

5. cv2.threshold(gray, 127, **?**, cv2.THRESH_BINARY)

    the value to assign to pixels above the threshold (white).

QUESTION 3: **[5 POINTS]**

The figure below shows the grayscale histogram of an image.



Using the histogram, answer the following:

1. Describe whether the image is mostly dark, mostly bright, or balanced.
   The image is balanced, but slightly leaning toward the brighter side because a large peak appears around intensity 200.
2. Identify the approximate intensity range where the highest peak occurs.
   The highest peak occurs around 180–210, which represents bright pixels.
3. Does the image contain good contrast or poor contrast? Explain your answer.
   The image has good contrast because the histogram spreads across a wide range of intensities (from near 0 to near 255), showing both dark and bright values.
4. Is there more detail in the dark regions or the bright regions? Explain using the histogram.
   There is more detail in the bright regions because the largest peak is in the high-intensity zone (≈200), meaning many pixels belong to brighter areas of the image.
5. Based on the histogram, does the image contain more dark pixels or more bright pixels? Explain how you can tell from the peak values.
   The image contains more bright pixels. This is clear because the highest peak in the histogram is located in the high-intensity range (around 180–200), which corresponds to bright areas in the image. The dark region (near 0–50) has a smaller peak, meaning fewer dark pixels.

QUESTION 4: [10 POINTS]

You have recently joined a technology company called ***VisionTech Solutions***, which specializes in building computer vision tools for medical, security, and industrial clients. Your manager assigns you a task to prepare a basic image-processing pipeline using Python, OpenCV, and Google Colab. A Google Drive folder named "UoP images" contains several color images that the company needs to process.

You are told that the folder path is: **"/content/drive/MyDrive/UoP images/"**

The company already provided you with a Python list called:

$$image\_files = ["img1.jpg", "img2.jpg", \dots]$$

which contains all the image names inside "UoP images".

Write a complete Python program that applies the following image processing pipeline to every image in "UoP images":

1. Access the folder path /content/drive/MyDrive/UoP images/.
2. Loop through every file name in image_files.
3. Read each image using cv2.imread().
4. Convert the image to grayscale.
5. Resize the grayscale image to 320×320 pixels.
6. Apply Gaussian Blur using kernel (5,5) and sigma = 0.
7. Apply Canny Edge Detection with thresholds 100 and 200.
8. Create a subfolder named "Processed" inside "UoP images" if it does not exist.
9. Save each processed image into "Processed" with a prefix "edge_".
10. Ensure that your program uses correct file paths, correct OpenCV functions, and proper indentation.

Write the full Python code that performs ALL the above steps. Your answer should be complete and syntactically correct.

```
import cv2
import os
 #Folder path in Google Drive
folder_path = "/content/drive/MyDrive/UoP images"/
 #The company already gave this list of filenames
 #image_files = ["img1.jpg", "img2.jpg", ... ]
 #Create the "Processed" folder path
processed_path = os.path.join(folder_path, "Processed")
 #Create "Processed" folder if it does not exist
if not os.path.exists(processed_path):
   os.makedirs(processed_path)
```

```
#Process each image in the list
for file_name in image_files:
.1 #    Read the image
   img_path = os.path.join(folder_path, file_name)
   img = cv2.imread(img_path)
#    Safety check
   if img is None:
      continue
.2 #    Convert to grayscale
   gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
.3 #    Resize to 320 × 320
   resized = cv2.resize(gray, (320, 320))
.4 #    Apply Gaussian Blur (5x5, sigma=0)
   blurred = cv2.GaussianBlur(resized, (5, 5), 0)
.5 #    Apply Canny Edge Detection(200 ,100)
   edges = cv2.Canny(blurred, 100, 200)
.6 #    Save in "Processed" folder using original name
   save_path = os.path.join(processed_path, file_name)
   cv2.imwrite(save_path, edges)
```

Good Luck