

<b>Petra University</b> جامعة البتراء		<b>25</b> جامعة البتراء - خمسة وعشرون عاماً University of Petra Anniversary (جامعة خاصة معتمدة)
<b>Private Accredited University</b>		
<b>Faculty of Information Technology</b>		كلية تكنولوجيا المعلومات

**Computer Vision(606384\_1)**  
**Midterm Exam**

Department: DSAI

Academic Year: 2025-2026

Term: 1<sup>st</sup>

Date: December 14, 2025

Instructor(s): Dr. Mohammad Al-Oudat

Coordinator: Dr. Mohammad AL-Oudat

Student's Name: Farah Mustafa

Student's ID: 202120337

Section: 1

Question	Points	Score
1	Out of 10	10
2	Out of 5	5
3	Out of 5	<del>5</del> 4
4	Out of 10	9
Total	30	28

**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:

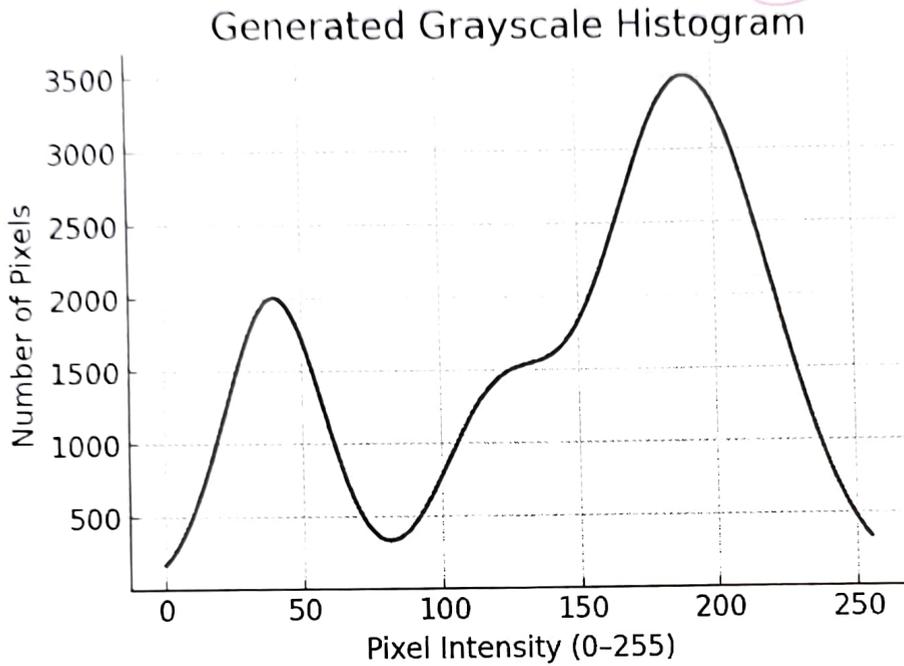
1. What is an image made of?
  - a. Only brightness values
  - b. A matrix of pixels
  - c. A list of text values
  - d. Audio samples
  
2. What does converting an image to grayscale do?
  - a. Adds more color
  - b. Makes the image 3-channel
  - c. Reduces it to one channel (0–255 values)
  - d. Converts 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



QUESTION 3:

The figure below shows the grayscale histogram of an image.

[5 POINTS]



Using the histogram, answer the following:

1. Describe whether the image is mostly dark, mostly bright, or balanced.
2. Identify the approximate intensity range where the highest peak occurs.
3. Does the image contain good contrast or poor contrast? Explain your answer.
4. Is there more detail in the dark regions or the bright regions? Explain using the histogram.
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.

1. the image is mostly bright, as there is a lot of pixels that ranges between  $[150, 255]$ , and somehow balanced in the middle.
2. the highest peak occurs from nearly  $[150$  to  $220]$ , means above 127 mostly bright
3. most pixels appear in a normal range of pixels intensity not too bright, but mostly bright, ~~so the~~ and some pixels are dark, but the bright ones is more. the image contains good contrast.
4. details appear most in bright, because it has more pixels, but if there is any dark detail it will be in the dark side.
5. it contains more bright pixels, as the peak is on the bright side from  $[150, 255]$ , which has peak here

#### 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", ...]
```

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 `320x320` 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.

Good Luck

from Google Colab import drive  
mount\_drive (%Content / Drive / My Drive) .

folder = %Content / Drive / My Drive / UOP Images / image\_files

~~from~~ import os

import cv2

from google.colab.patches import ~~imshow~~ cv2\_imshow.

supported = ['.jpg', '.jpeg', '.bmp', '.png']

files = [f for f in os.listdir(folder) if f.lower().endswith(supported)]

for fname in files:

img\_path = os.path.join(fname, files)

img = cv2.imread(img\_path)

cv2\_imshow(img)

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

gray\_resize = cv2.resize(img, (320, 320))

Gussian = cv2.GaussianBlur(img, (5, 5), 0)

Canny = cv2.Canny(<sub>cv2</sub>img, cv\_64F, 100, 200)

cv2\_imshow(img)

→ create subfolder here

\* Save

cv2.imwrite("edge.jpg", img)