

Petra University		 جامعة البتراء - خمسة وعشرون عاماً 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: Maysa Shehben

Student's ID: 202210831

Section: 1

Question	Points	Score
1	Out of 10	9
2	Out of 5	1
3	Out of 5	4
4	Out of 10	5
Total	30	19

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 (computes)
 - 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 Parameter (?) with the correct term, and describe the purpose of it:

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

100, you're giving the parameters of the rectangle in this case
100 x 100 and 410 by 410
(width and height.)

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

(300, 300) half the equator, full equator.

3. `cv2.putText(img, "Hello", (60,40), cv2.FONT_HERSHEY_SIMPLEX, 1, ?, 2)`

you input a text size for example (17)

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

you put the intended axis either (x-axis) or (y-axis) the is meant to be scaled

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

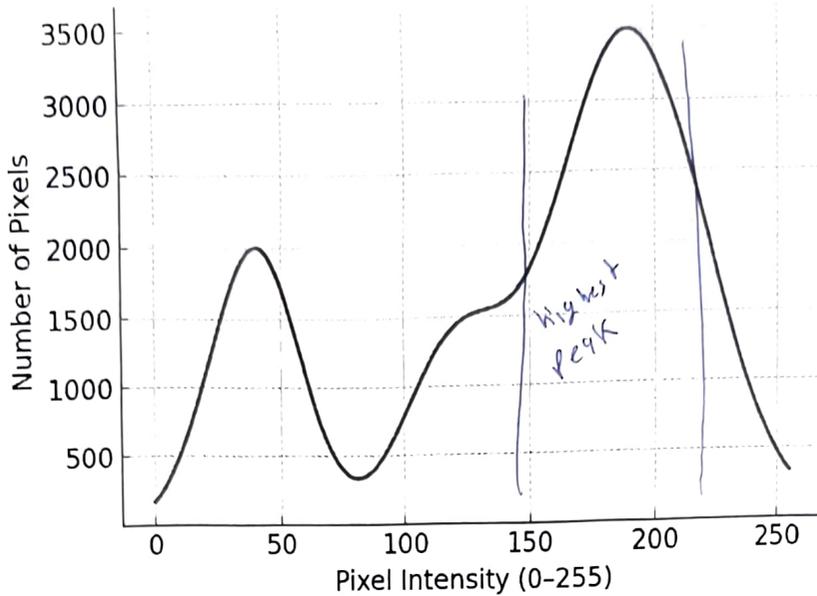
255, after putting the threshold (127) we put the value of the highest greyscale value in which it's shown as the color white.

QUESTION 3:

[5 POINTS]

The figure below shows the grayscale histogram of an image.

Generated Grayscale Histogram



Using the histogram, answer the following:

1. Describe whether the image is mostly dark, mostly bright, or balanced.

Mostly Bright, because the highest peaks of the image occurs in the (150-200+) range.

2. Identify the approximate intensity range where the highest peak occurs.

(150 - 220) approximately.

3. Does the image contain good contrast or poor contrast? Explain your answer.

Poor contrast, because most of the intensity is in the higher regions of the pixels (right side) and there isn't much balance in the darker/gray areas.

4. Is there more detail in the dark regions or the bright regions? Explain using the histogram.

The bright regions because the more number of pixels are the more neighbors they have which helps in the detail and the contrast between them.

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.

More bright pixels, Because the peak in the higher pixels is larger which are the white color and number of pixels are higher in the brighter region.

[10 POINTS]

QUESTION 4:

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

Q. 4)

```

import google drive (/content/drive/My drive/Vop images)
mount drive
import matplotlib as plt
import os
import cv2

```

```

for z cv2.open (image_files) z

```

```

open if cv2.open == img1.jpg
    then cv2.open cv2.open (image_files) ?

```

```

img = cv2.imread ("/content/drive/My Drive/Vop images/")

```

```

gray = cv2.cvtColor (img, cv2.COLOR_BGR2GRAY) (img, (0, 255))

```

```

resize = cv2.resize (gray, (320, 320))

```

```

blur = cv2.Blur (resize, kernel = 5, 0, cv2.gaussianBlur)

```

```

canny = cv2.canny (blur, lowerThresh = 100, higherThresh = 200, cv2.canny cannyThresh)

```

```

if (cv2.open ("processed", "/content/drive/My drive/Vop images/"))

```

== not successful

```

    then cv2.create ("processed", "/content/drive/My Drive/Vop images")

```

else

```

cv2 cv2.open ("processed")

```

}

```

cv2.save ("processed", edge = "blur")

```

```

cv2.write ("/content/drive/My drive/Vop images")

```

```

if cv2.check == successful

```