Logo, company name

Description automatically generated

Digital image Processing

**Assignment No 1**

**Report**

**Group Members:**

**Ghulam Hussain Khan Sherwani 366549**

**Abdul Basit Sajid 386804**

**Mohammad Irfan 373668**

Due Date: 27 - 09 - 2023

Submission Date: 27 - 09 - 2023

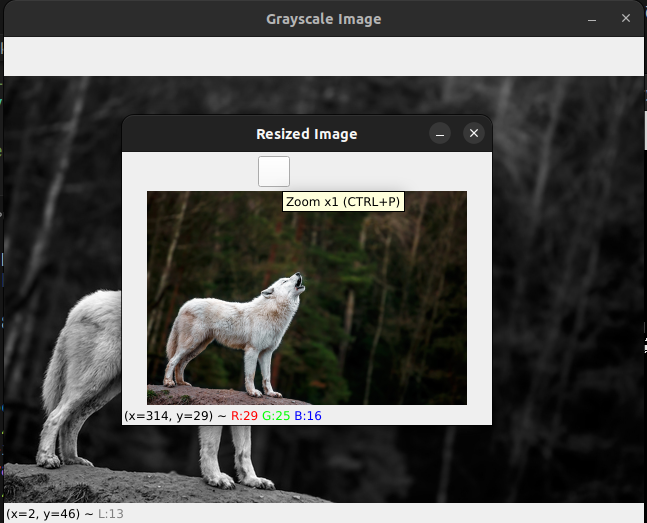
Submitted to: Muhammad Asad Khan

# Original Image:

# Task1:

## Resized Image:

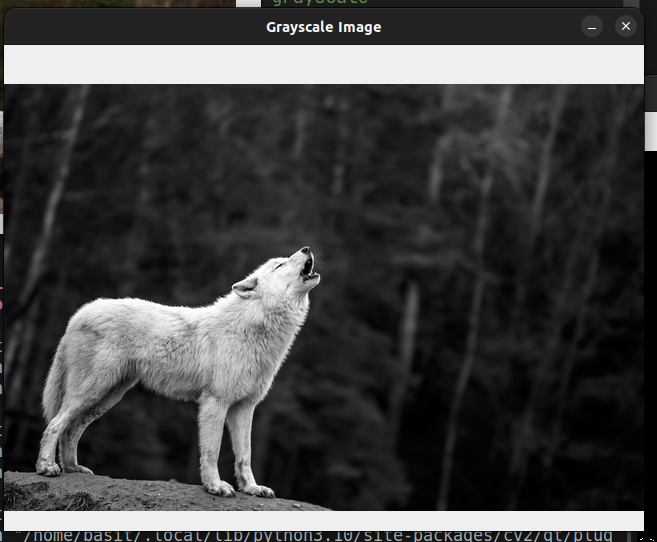
Resized down image compared to the original size image (but in greyscale)



## 

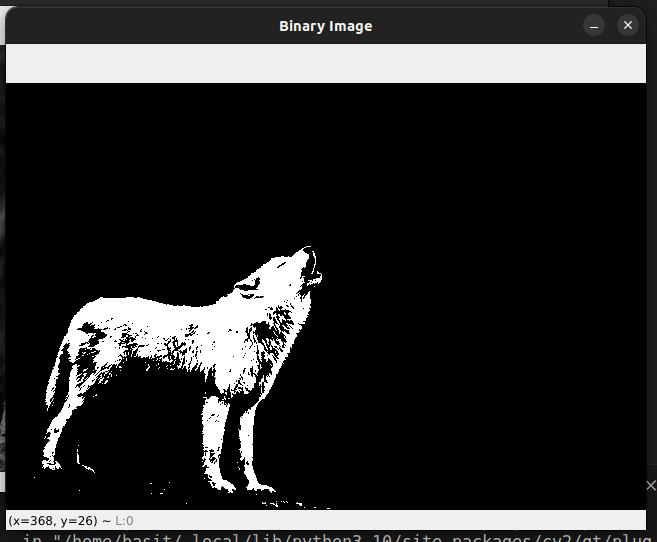
# Task2:

## Gray Scale:



# Task3:

## Binary Image:



## Combined Code for above 3 tasks:

import cv2

# Load the RGB image

image = cv2.imread('./wolf.jpg')

# Convert the image to grayscale

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Threshold the grayscale image to create a binary image !! fine-tuned the threshold

\_, binary\_image = cv2.threshold(gray\_image, 148, 255, cv2.THRESH\_BINARY)

scaling\_factor = 0.5 # Change this to own desired factor

# Resize the image while maintaining the aspect ratio

resized\_image = cv2.resize(image, None, fx=scaling\_factor, fy=scaling\_factor)

# Save the resized image

cv2.imwrite('output.jpg', resized\_image)

# Display the resized image (optional)

cv2.imshow('Resized Image', resized\_image)

# Save the grayscale image

cv2.imwrite('gray\_image.jpg', gray\_image)

# Save the binary image

cv2.imwrite('binary\_image.jpg', binary\_image)

# Display the grayscale and binary images (optional)

cv2.imshow('Grayscale Image', gray\_image)

cv2.imshow('Binary Image', binary\_image)

# Wait for a key press and then close the windows

key=cv2.waitKey(0)

while key!=ord(" "):

key=cv2.waitKey(0)

cv2.destroyAllWindows()

# Task4:

## Count Coins:

No. of Coins in the image: 8



## Code:

import cv2

import numpy as np

image = cv2.imread('./coins.png')

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

#blur the image

blurred = cv2.GaussianBlur(gray\_image, (15, 15), 0)

threshold\_value = 155

# Apply binary thresholding

ret, binary\_image = cv2.threshold(blurred, threshold\_value, 255, cv2.THRESH\_BINARY)

#invert the image

inverted\_image = cv2.bitwise\_not(binary\_image)

# Find connected components in the inverted image

connectivity = 0 #

output = cv2.connectedComponentsWithStats(inverted\_image, connectivity, cv2.CV\_32S)

# Extract the connected components and their statistics

num\_labels = output[0]

labels = output[1]

# Count the number of connected dark regions (islands)

num\_of\_coins = num\_labels - 1 # Subtract 1 to exclude the background label

print(f"Number of coins: {num\_of\_coins}")

## Github Link:

<https://github.com/abd-ulbasit/digital-image-processing/tree/main/assignments/first>