

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту



Звіт з лабораторної роботи №1
з курсу “Теорія інформації”

Виконав:
ст. гр. КН-210
Сагайдак Н. І.

Викладач:
Косаревич Р. Я.

Львів – 2019

Мета роботи: вивчення властивостей ентропії як кількісної міри інформації.

Завдання

Варіант - 23

1. Знайти ентропію дискретної випадкової величини X , заданої розподілом $P(x_i)$. Значення $P(x_i)$ взяти згідно варіанту.

$p(x_1)$	$p(x_2)$	$p(x_3)$	$p(x_4)$	$p(x_5)$	$p(x_6)$	$p(x_7)$	$p(x_8)$	$p(x_9)$	$p(x_{10})$
0.09	0.12	0.09	0.12	0.09	0.07	0.13	0.11	0.07	0.11

2. Розробити програму для визначення ентропії зображення. В програмі передбачити наступні функції:

- читання файлу зображення;
- підрахунок загальної кількості пікселів зображення;
- групування пікселів за інтенсивністю свідчення у 16 груп;
- відображення ймовірностей значень інтенсивності свідчення груп пікселів у блоках;
- розрахунок ентропії з інформації зображення за формулою;
- розрахунок ентропії з використанням вбудованої функції;
- відображення обчисленого значення ентропії.

3. За допомогою програми визначити ентропію для одного і того ж зображення збереженого у форматах bmp, jpeg, tiff, png.

4. Провести порівняльну оцінку одержаних результатів.

Виконання

1. Обчислюємо за формулою:

$$H(X) = - \sum_{i=1}^{10} P(x_i) * \log(P(x_i)) = 2,535$$

2.1. Код програми мовою python:

```
from matplotlib import pyplot
from PIL import Image
import math
import cv2

def histogram_of_image(path):
    img = cv2.imread(path, 0)
    histogram = cv2.calcHist([img], [0], None, [256], [0, 256])
    print("Histogram of " + path[path.rfind("/") + 1:] + ".")
    pyplot.plot(histogram)
    pyplot.show()
```

```

def count_of_pixels(image, path):
    width, height = image.size
    print("Count of pixels in " + path[path.rfind("/") + 1:] + ": " + str(width * height) + ".")

def open_image(path):
    try:
        image = Image.open(path)
        return image.convert("L")
    except IOError:
        print("File cannot be found, or image cannot be opened.")

def split_list(alist, wanted_parts=1):
    length = len(alist)
    return [alist[i*length // wanted_parts: (i+1)*length // wanted_parts] for i in
range(wanted_parts)]

def entropy(image, path):
    histogram = image.histogram()
    sum_of_histogram = sum(histogram)
    histogram = [i / sum_of_histogram for i in histogram]
    result = -sum([i * math.log(i, 2) for i in histogram if i != 0])
    print("Entropy of " + path[path.rfind("/") + 1:] + ": " + str(result) + ".")

def samples_probability_in_blocks(list):
    for i in range(len(list)):
        sum_of_block = sum(list[i])
        if sum_of_block == 0:
            continue
        for j in range(len(list[i])):
            list[i][j] /= sum_of_block
    return list

def print_blocks(list):
    for row in list:
        print(' '.join([str(elem) for elem in row]))

path_for_jpeg, path_for_png, path_for_tiff = "/home/master/Desktop/душить
питона/TI/lab1/lab1.jpeg", \
"/home/master/Desktop/душить питона/TI/lab1/lab1.png", \
"/home/master/Desktop/душить питона/TI/lab1/lab1.tiff"
image_jpeg = open_image(path_for_jpeg)
image_png = open_image(path_for_png)
image_tiff = open_image(path_for_tiff)

```

```

print_blocks(samples_probability_in_blocks(split_list(image_jpeg.histogram(),
wanted_parts=16)))
count_of_pixels(image_jpeg, path_for_jpeg)
entropy(image_jpeg, path_for_jpeg)
entropy(image_png, path_for_png)
entropy(image_tiff, path_for_tiff)
histogram_of_image(path_for_jpeg)
histogram_of_image(path_for_png)
histogram_of_image(path_for_tiff)
image_jpeg.close()
image_png.close()
image_tiff.close()

```

2.2 Код програми для обчислення ентропії вбудованою функцією:

```

entropy_of_jpeg = entropy(imread("lab1.png"))
entropy_of_png = entropy(imread("lab1.png"))
entropy_of_tiff = entropy(imread("lab1.tiff"))
printf("Entropy of lab1.jpeg: %d.\n", entropy_of_jpeg)
printf("Entropy of lab1.png: %d.\n", entropy_of_png)
printf("Entropy of lab1.tiff: %d.\n", entropy_of_tiff)

```

Результати виконання програм:

```

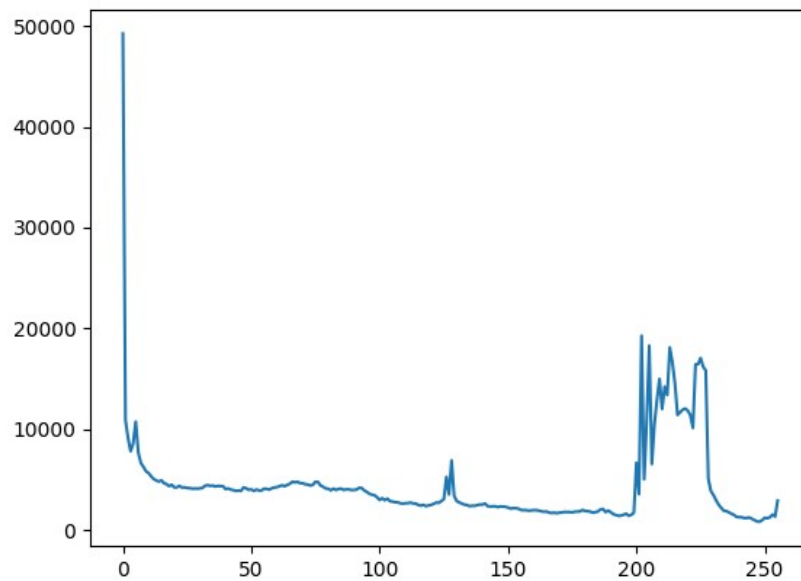
Samples probability in blocks
0.23041643617089708 0.11400856801018898 0.08891561925100668 0.06432440081820638 0.06357823776871518 0.07010716445176313 0.048880
0.06730599776772601 0.06547024613757857 0.06511778182459026 0.065014979733302 0.06151970862950126 0.06273864771191917 0.06276801
0.06728826496250954 0.06522292231019051 0.06495352979032282 0.06445964350389871 0.06420521723513477 0.06589640360985977 0.065207
0.06060375983235643 0.06171227260303095 0.05948006195523431 0.059768578977738636 0.06081635132262277 0.059191544932729975 0.0614
0.062323609179894546 0.06513209521318336 0.06533860153916048 0.06488428762201082 0.06620592810826438 0.06297066233462285 0.06370
0.06204996556258218 0.06286394089286833 0.06327092855801139 0.06187777847348319 0.063881410055726 0.06305178135370358 0.06317700
0.07561829954357287 0.07413225772211018 0.0695042989066978 0.0661288610550897 0.06396348582952978 0.06404840250504193 0.06421823
0.05375643713488978 0.05446884732031997 0.05153778827169289 0.04856602006961265 0.05074395977935639 0.049848358403387 0.04817928
0.08184038103527477 0.06970334372960035 0.06902637750537946 0.06416672711007955 0.06481951596914969 0.062184183167718385 0.05749
0.06983272897414902 0.06649900573166452 0.06696689671306585 0.0662065738682887 0.06869224470698328 0.06591414200491286 0.0635454
0.06917731292994525 0.06812438579250316 0.06650989751509195 0.062087603537835184 0.06292994524778886 0.06282465253404465 0.05836
0.06307876302750726 0.0633862976251495 0.07069878694686485 0.06499231163505895 0.06232701178882624 0.06085767982231334 0.0603792
0.014145590922204057 0.01340513510914511 0.01398211366477546 0.01545340898163285 0.014020578901817482 0.013616693912876237 0.015
0.05744060542149991 0.05409093728676881 0.08155153574172246 0.061024082300678045 0.06479842346103744 0.07639342854279893 0.08396
0.2520160376931092 0.15563131427535903 0.20211344176142798 0.09216010510578534 0.049664748788112174 0.03367235989670638 0.031724
0.05978462681024879 0.06418757625590155 0.06498329001113999 0.05628348628719962 0.06530157551323537 0.05336586918465864 0.044347
Count of pixels in lab1.jpeg: 1143000.
Entropy of lab1.jpeg: 7.559059660422384.
Entropy of lab1.png: 7.468328912150118.
Entropy of lab1.tiff: 7.470236089877993.
Histogram of lab1.jpeg.

```

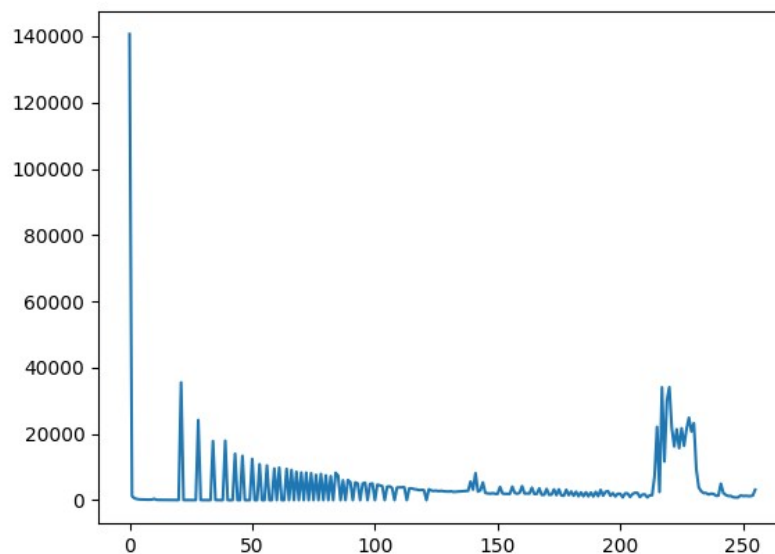
Entropy of lab1.jpeg: 7.35488.
Entropy of lab1.png: 7.15251.
Entropy of lab1.tiff: 7.15271.

3. Гістограми пікселів зображень:

3.1 lab1.jpeg

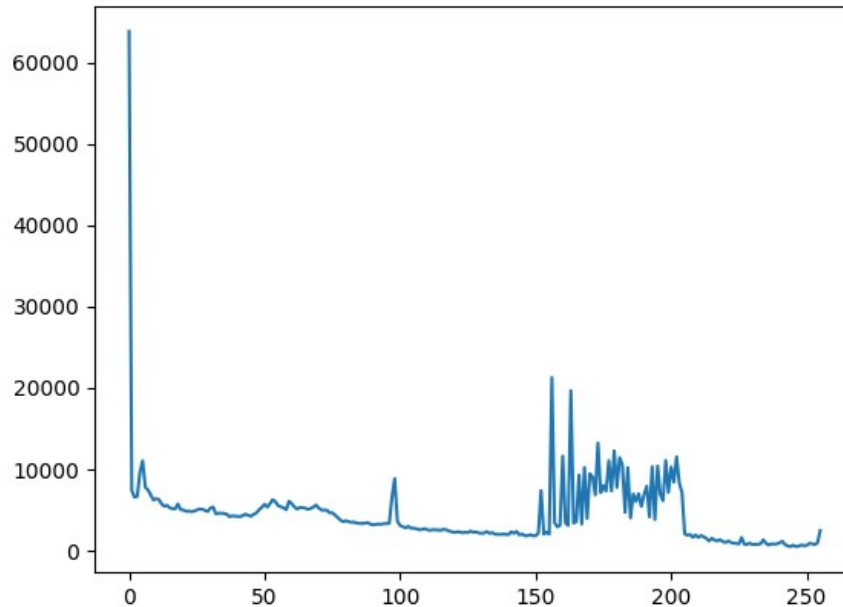


3.2 lab1.png

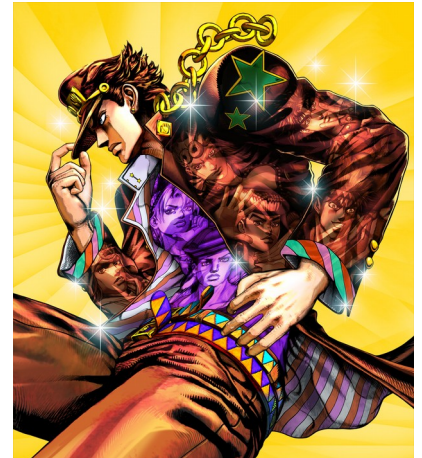


Гістограма для зображень lab1.png є некоректною, оскільки libpng не підтримує sRGB профілі старого покоління.

3.3 lab1.tiff



4. Зображення, що використовувалося для обчислення ентропії: lab1.jpeg, lab1.png, lab1.tiff.



5. Всі файли і програми до лабораторної роботи знаходяться за посиланням:
<https://github.com/NazariiSahaidak/Study/tree/master/TI/Lab1>

Висновок: виконуючи лабораторну роботу, я ознайомився з властивостями ентропії як кількісної міри інформації та навчився базовим методам дослідження кількості й різноманіття відтінків зображення.