МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. І.СІКОРСЬКОГО»

ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра фізико-технічних засобів захисту інформації

Лабораторна робота № 1

з дисципліни: «Автоматизація обробки ІзОД»

Керівник: Виконав:

Прогонов Дмитро Олександрович студент 5 курсу групи ФЕ-91мп

Захищено з оцінкою Павлусь Олександр Сергійович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, підпис

Київ – 2020 р.

Завдання лабораторної роботи:

1. Сформувати тестову вибірку зображень з вихідного пакета;
2. Для кожного каналу кольору кожного зображення з тестового пакета обчислити наступні характеристики:
   1. Максимальна / мінімальне значення;
   2. Математичне сподівання і дисперсію;
   3. Медіану значень, інтерквартільний розмах;
   4. Коефіцієнти асиметрії та ексцесу (нормалізований);
3. Для кожного каналу кольору кожного зображення з тестового пакета побудувати гістограму значень яскравості пікселів;
4. Провести апроксимацію отриманих гістограм з використанням відомих імовірнісних розподілів, визначити найкращу апроксимацію;
5. Побудувати розподіл типів використаних імовірнісних розподілів для яких досягається мінімальне значення помилки апроксимації з п.4.

1) Формування тестової вибірки зображеннь

Тестова вибірка сформована із псевдовипадкового вибору 250 кольорових зображень із пакету Mirflick. Розрахунки проводились за допомогою мови програмування Python 3.

Приклад коду для вибірки зображень:  
*import random*

*dataset = random.sample(range(25000), 250)*

*for img in dataset:*

*print(img)*

2) Обчислення характеристик для кожного каналу зображень

**Picture N7096**

**Для Red каналу:**

Мат. сподівання: 34.5550

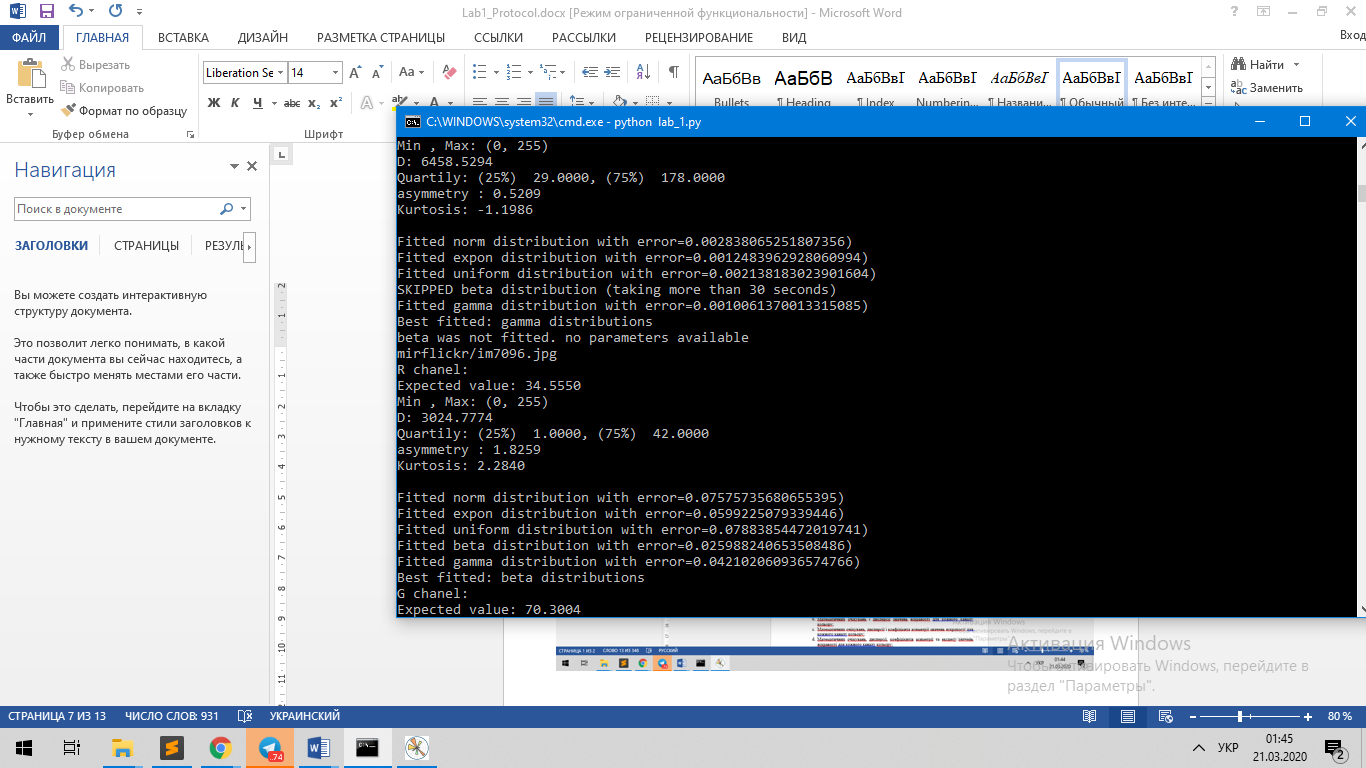
Мінімальне та максимальне значення: (0, 255)

Дисперсія: 3024.7774

Інтерквартільний розмах: (25%) = 1.0000, (75%) = 42.0000

Коефіцієнт асиметрії: 1.8259

Коефіцієнт ексцесу: 2.2840



**Для Green каналу:**

Мат. сподівання: 70.3004

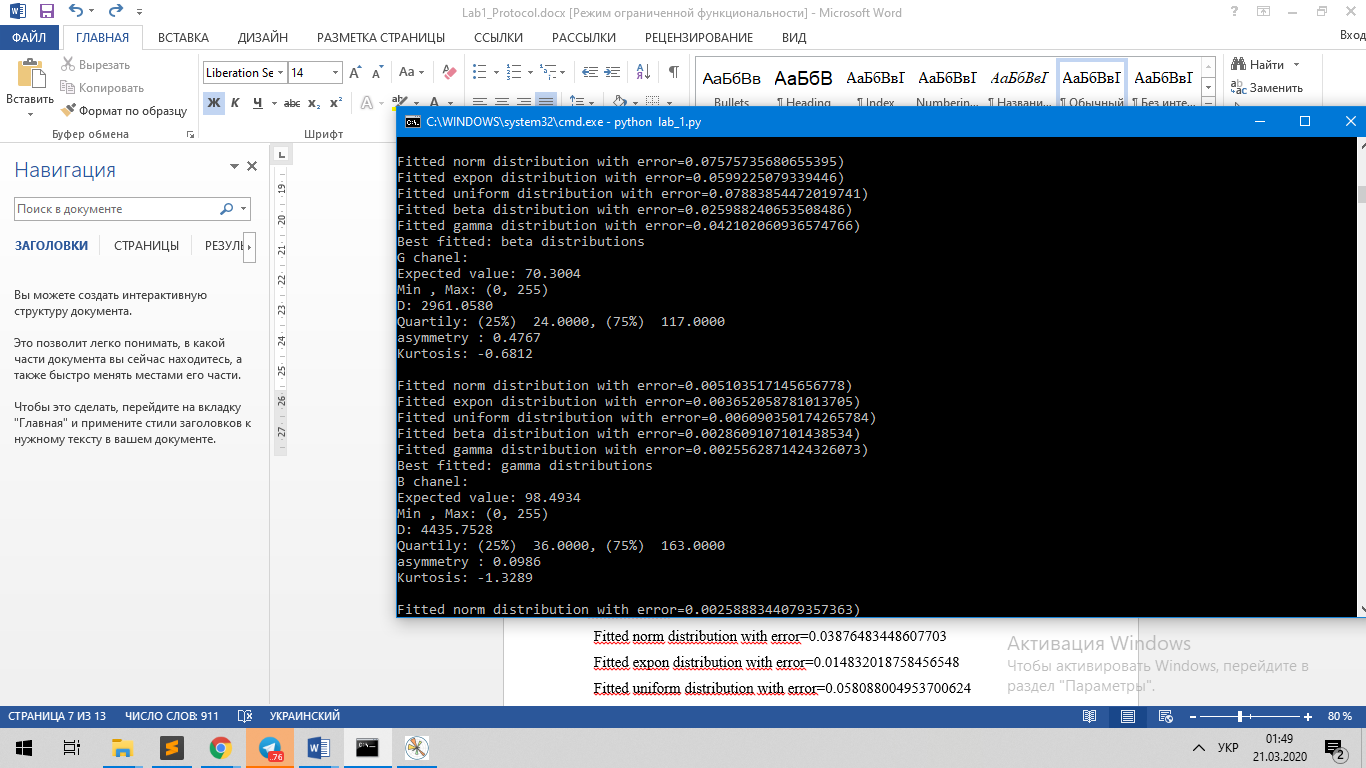
Мінімальне та максимальне значення: (0, 255)

Дисперсія: 2961.0580

Інтерквартільний розмах: (25%) = 24.0000, (75%) = 117.0000

Коефіцієнт асиметрії: 0.4767

Коефіцієнт ексцесу: -0.6812



**Для Blue каналу:**

Мат. сподівання: 98.4934

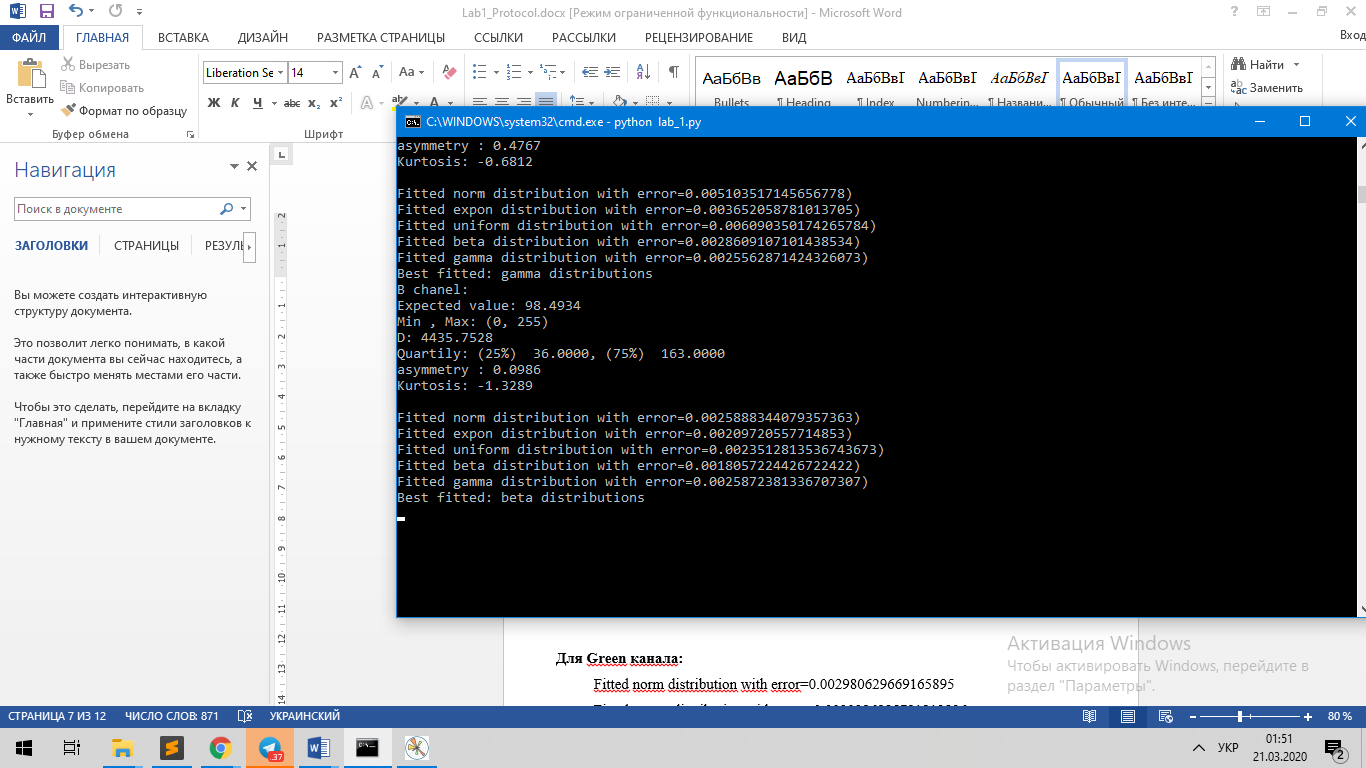
Мінімальне та максимальне значення: (0, 255)

Дисперсія: 4435.7528

Інтерквартільний розмах: (25%) = 36.0000, (75%) = 163.0000

Коефіцієнт асиметрії: 0.0986

Коефіцієнт ексцесу: -1.3289



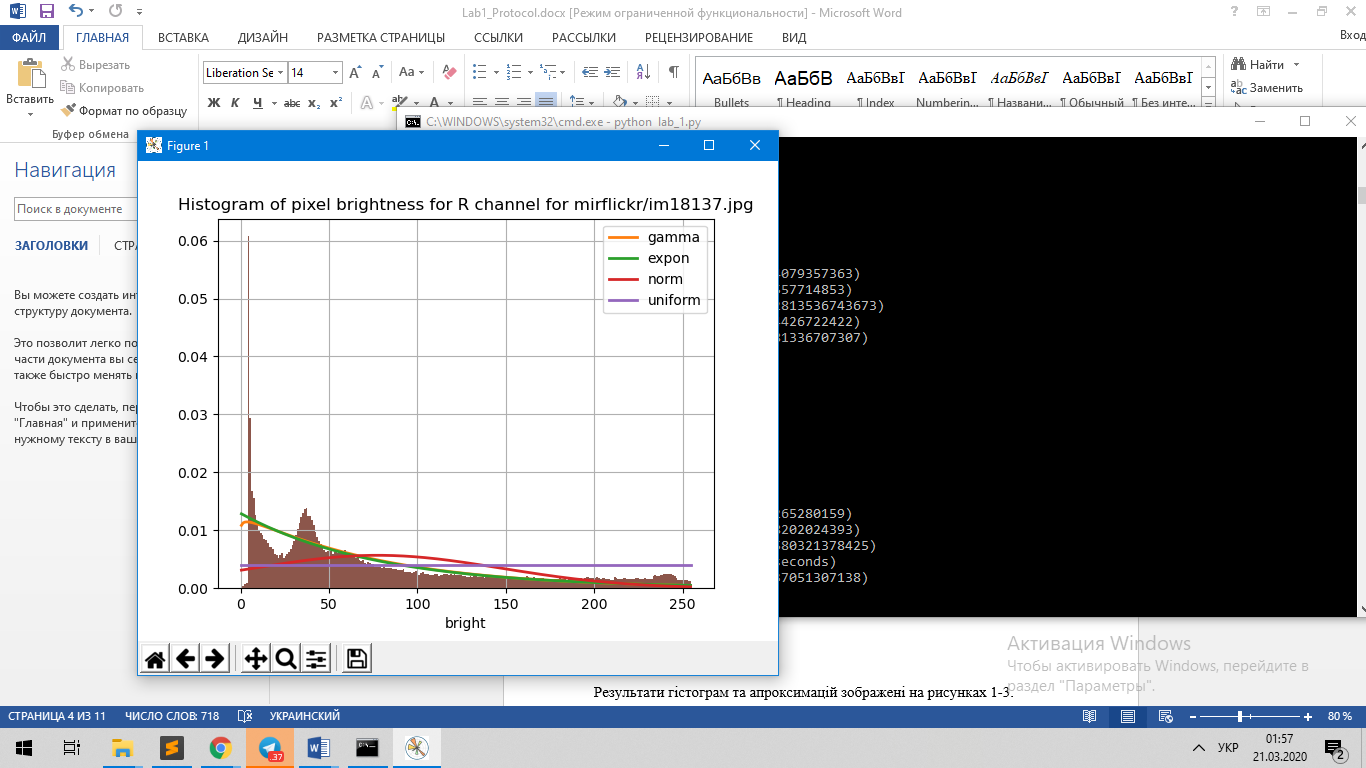
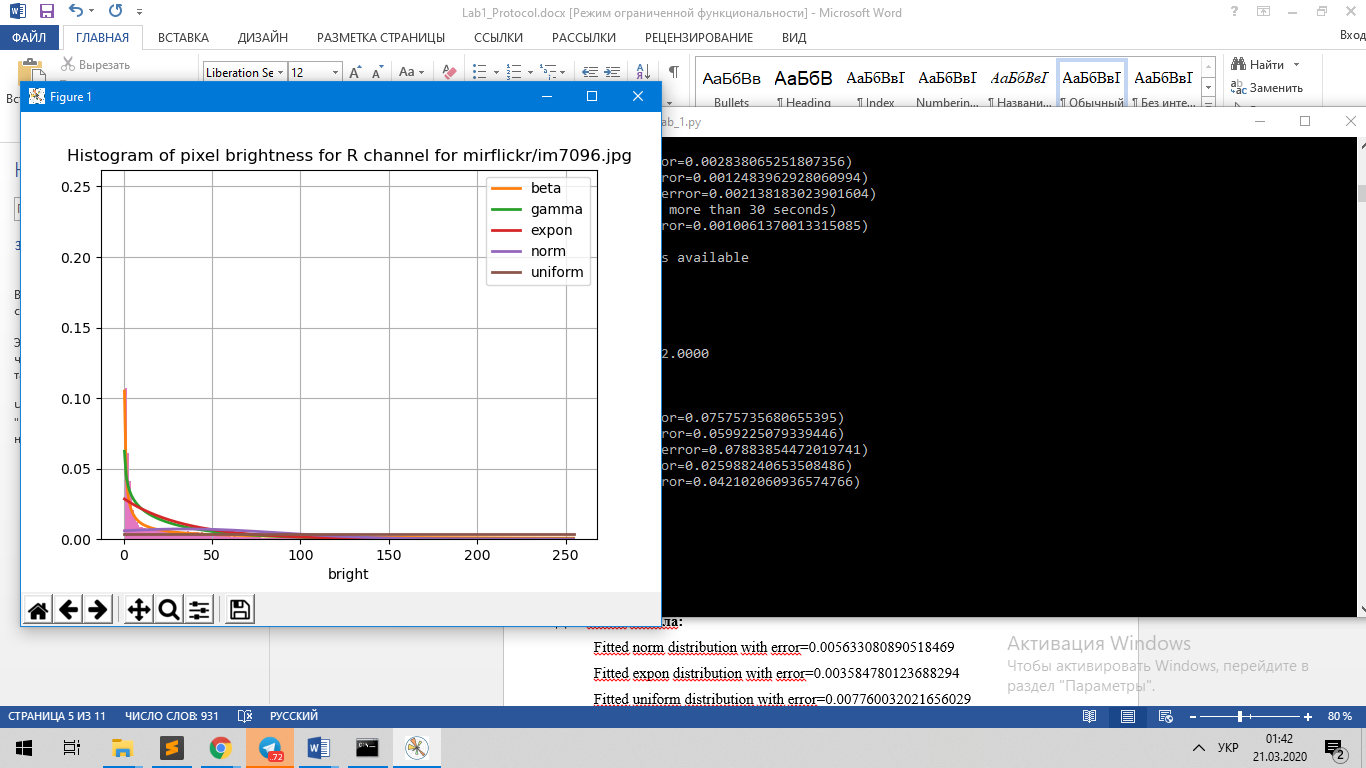
3) Для кожного каналу кольору кожного зображення з тестового пакета побудувати гістограму значень яскравості пікселів та провести апроксимацію отриманих гістограм з використанням відомих імовірнісних розподілів, визначити найкращу апроксимацію;

В роботі були використані наступні функції розподілу випадкової величини:

* Бета-розподіл
* Рівномірний розподіл
* Експоненціальний розподіл
* Нормальний розподіл

Результати гістограм та апроксимацій зображені на рисунках 1-3.

**1. Для Red каналу:**



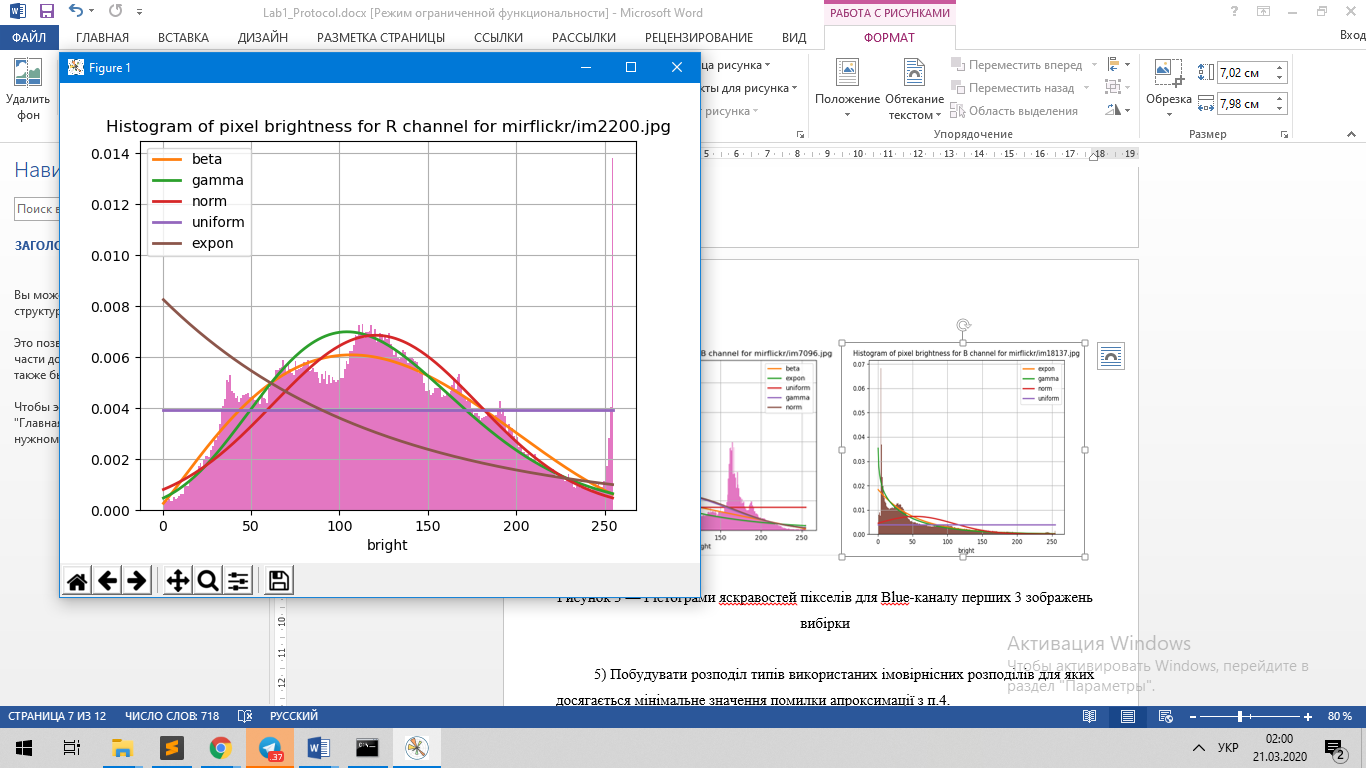
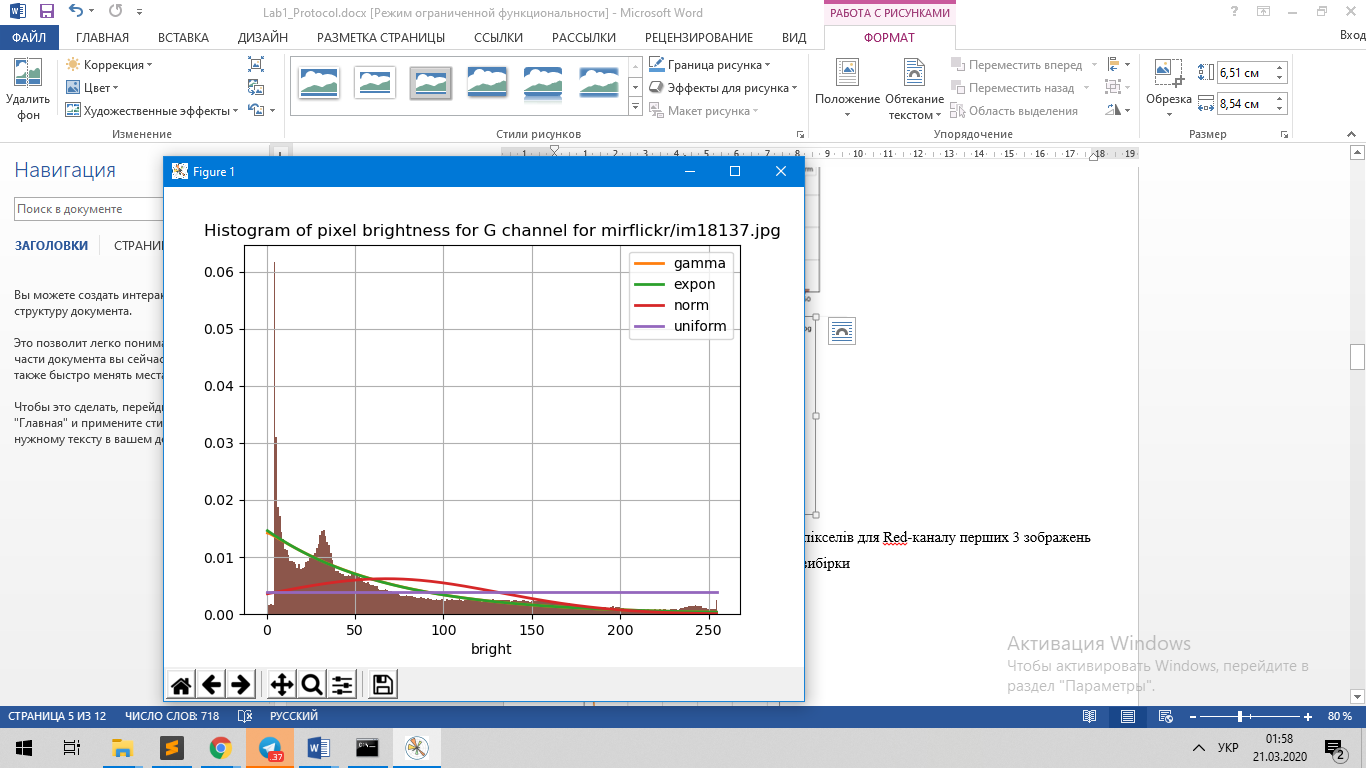
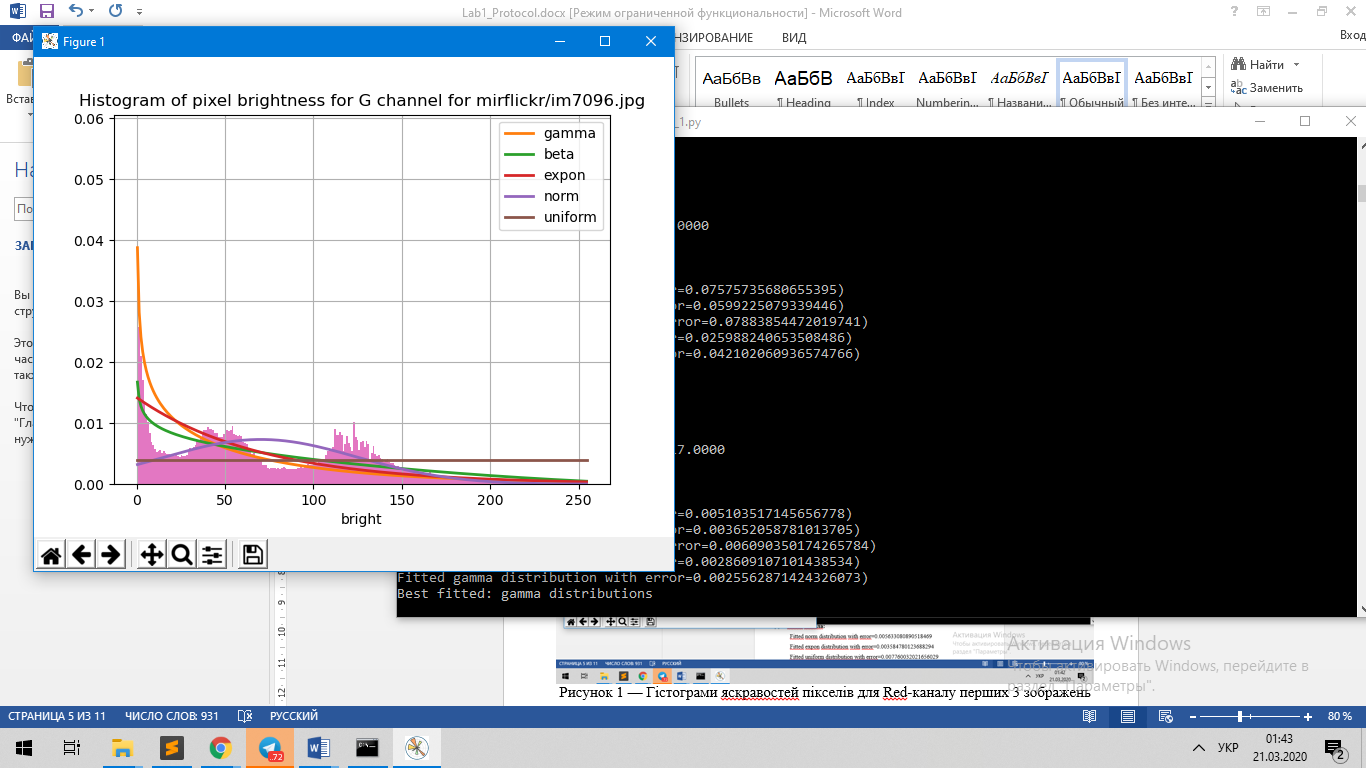


Рисунок 1 — Гістограми яскравостей пікселів для Red-каналу перших 3 зображень вибірки

**2. Для Green каналу:**



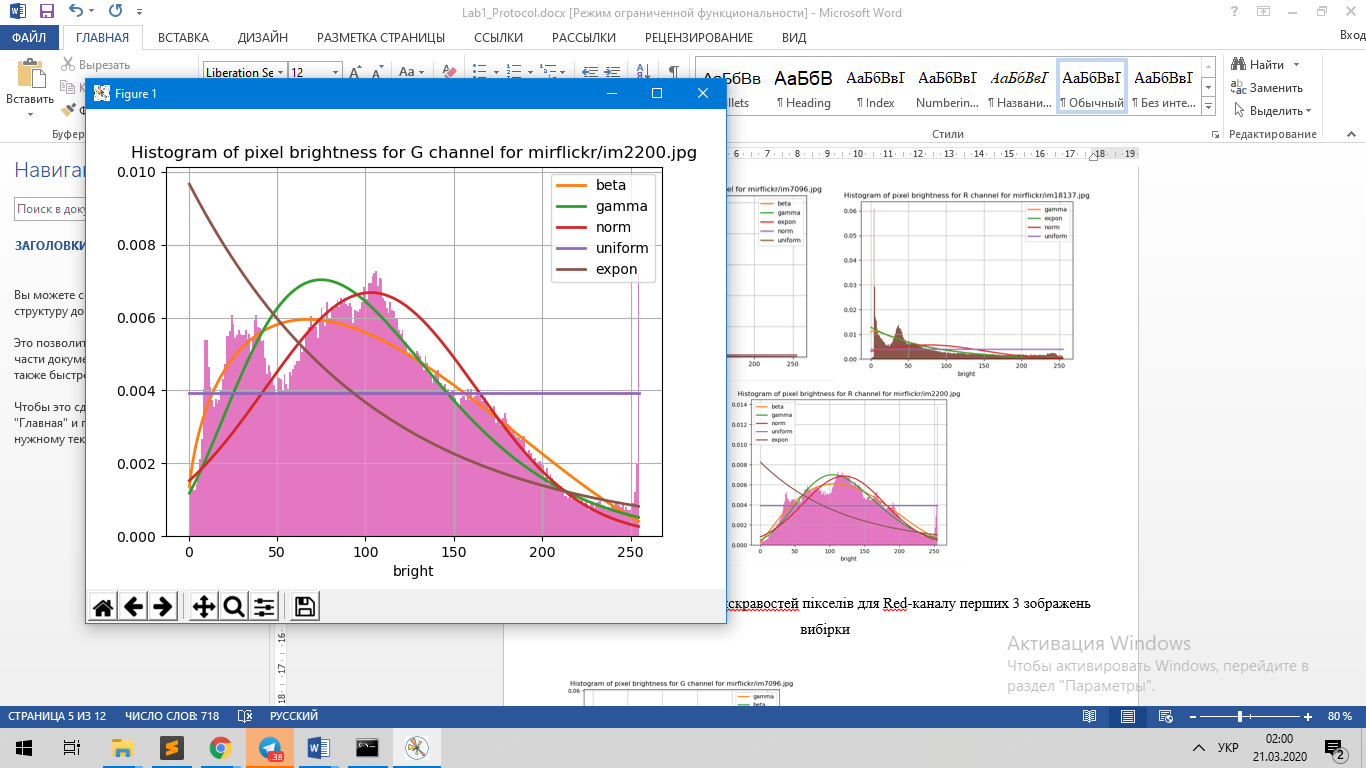
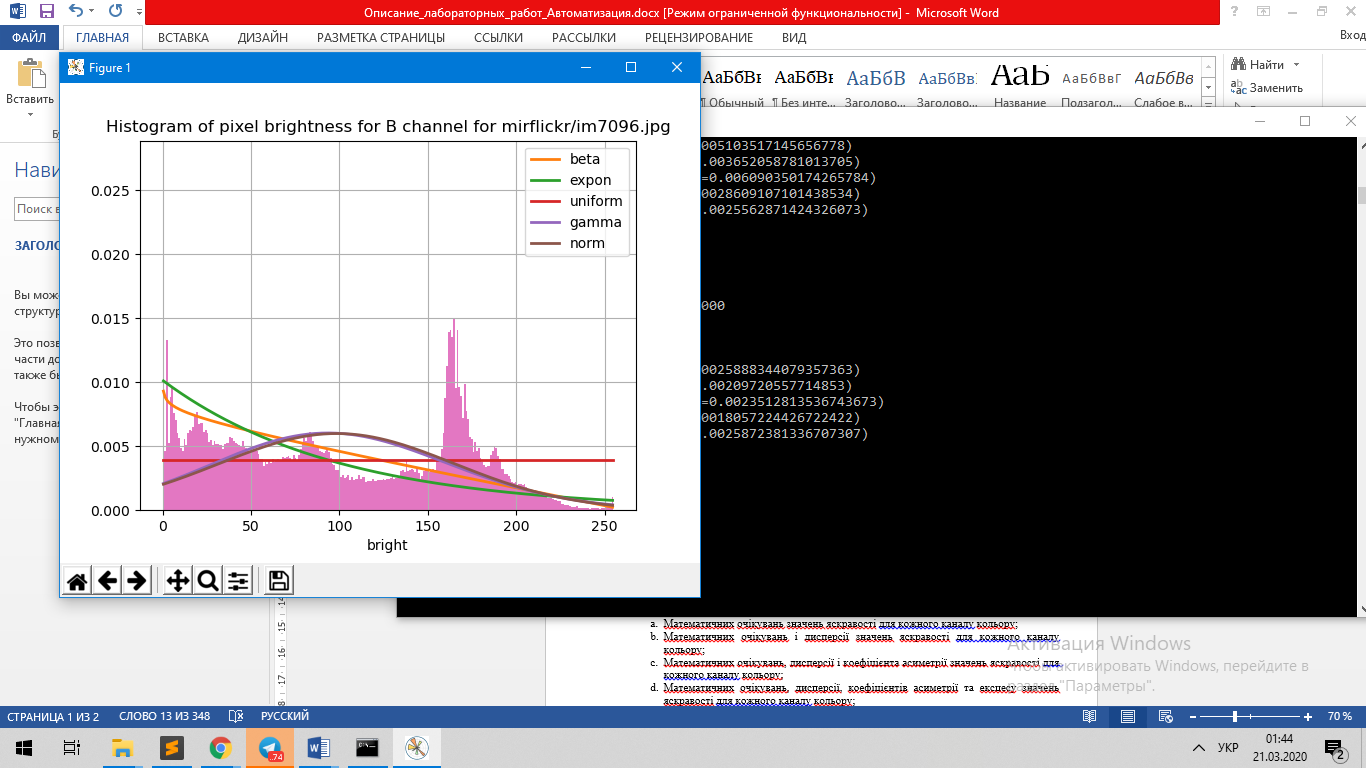
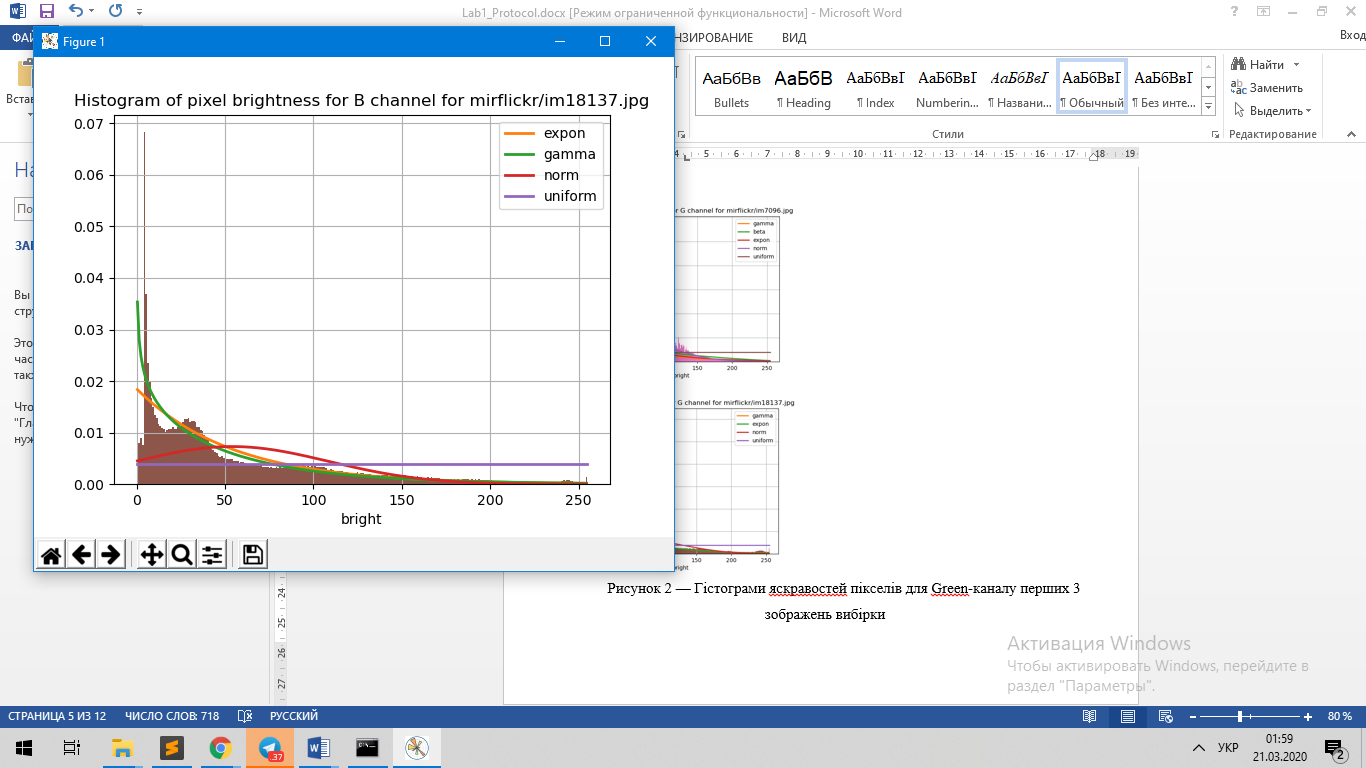


Рисунок 2 — Гістограми яскравостей пікселів для Green-каналу перших 3 зображень вибірки

**3. Для Blue каналу:**

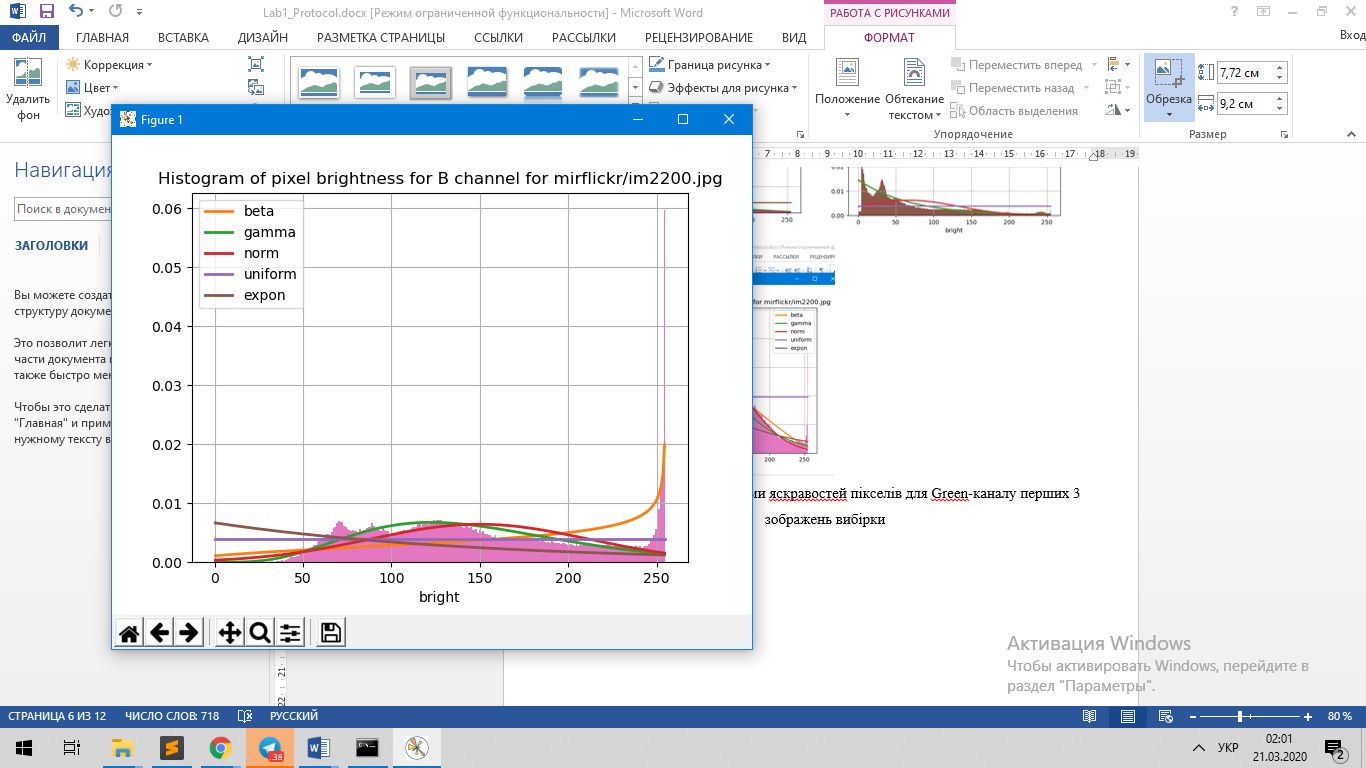


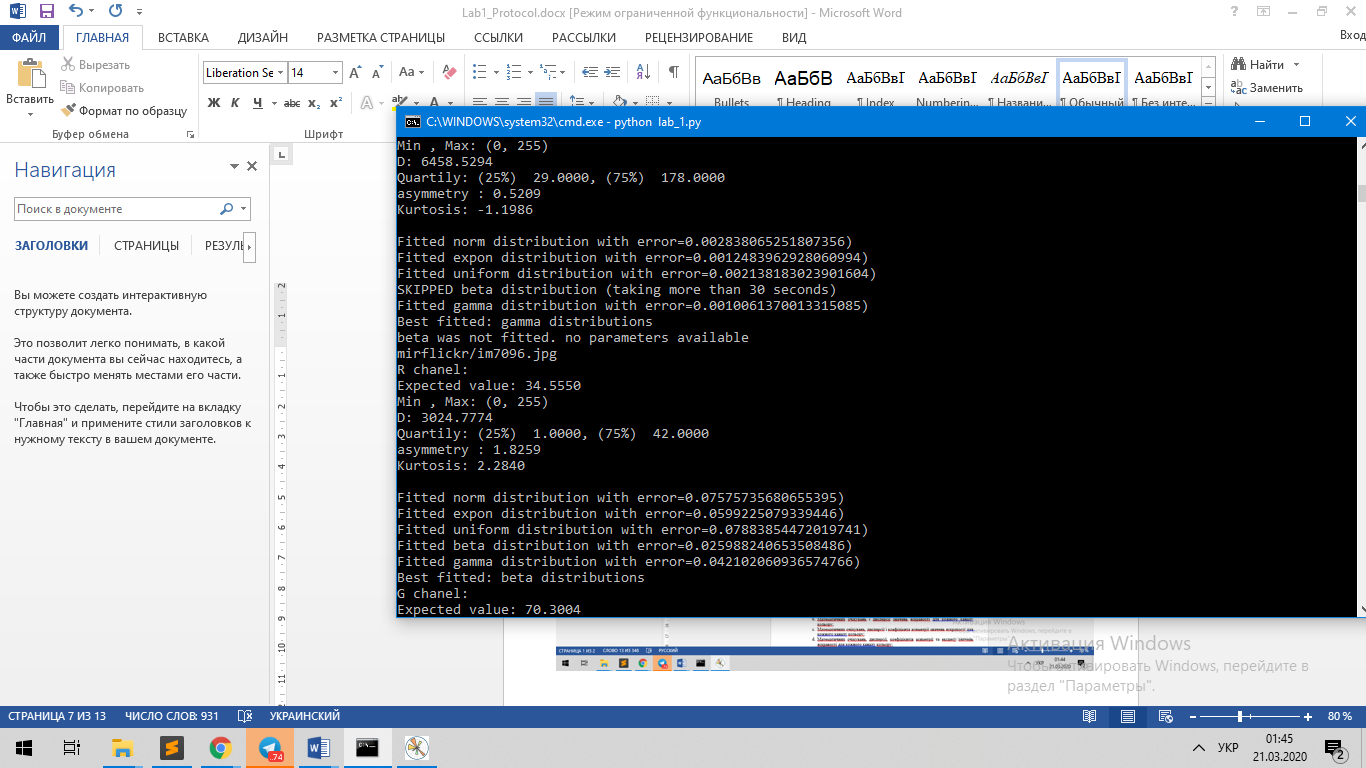
Рисунок 3 — Гістограми яскравостей пікселів для Blue-каналу перших 3 зображень вибірки

4) Побудувати розподіл типів використаних імовірнісних розподілів для яких досягається мінімальне значення помилки апроксимації з п.4.

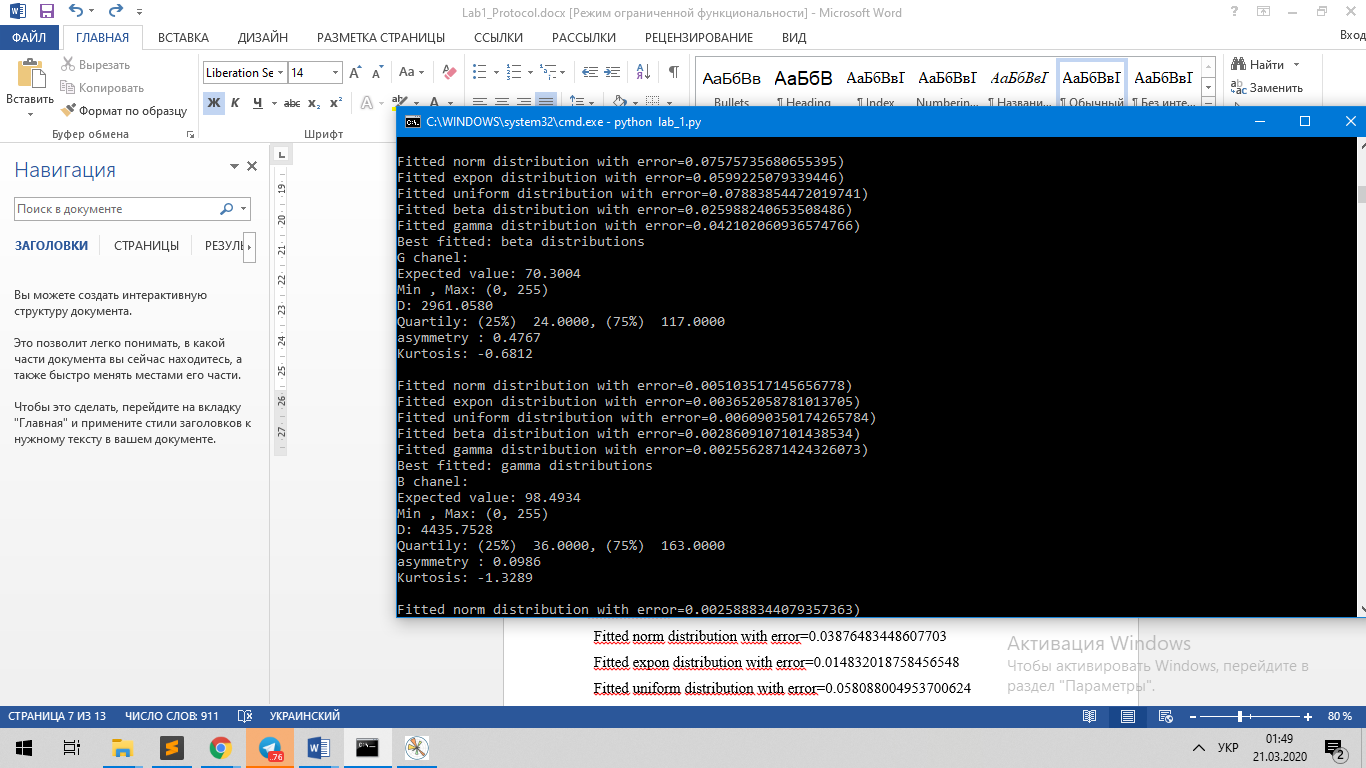
Для зображення покажемо похибку апроксимації:

**1. picture N7096**

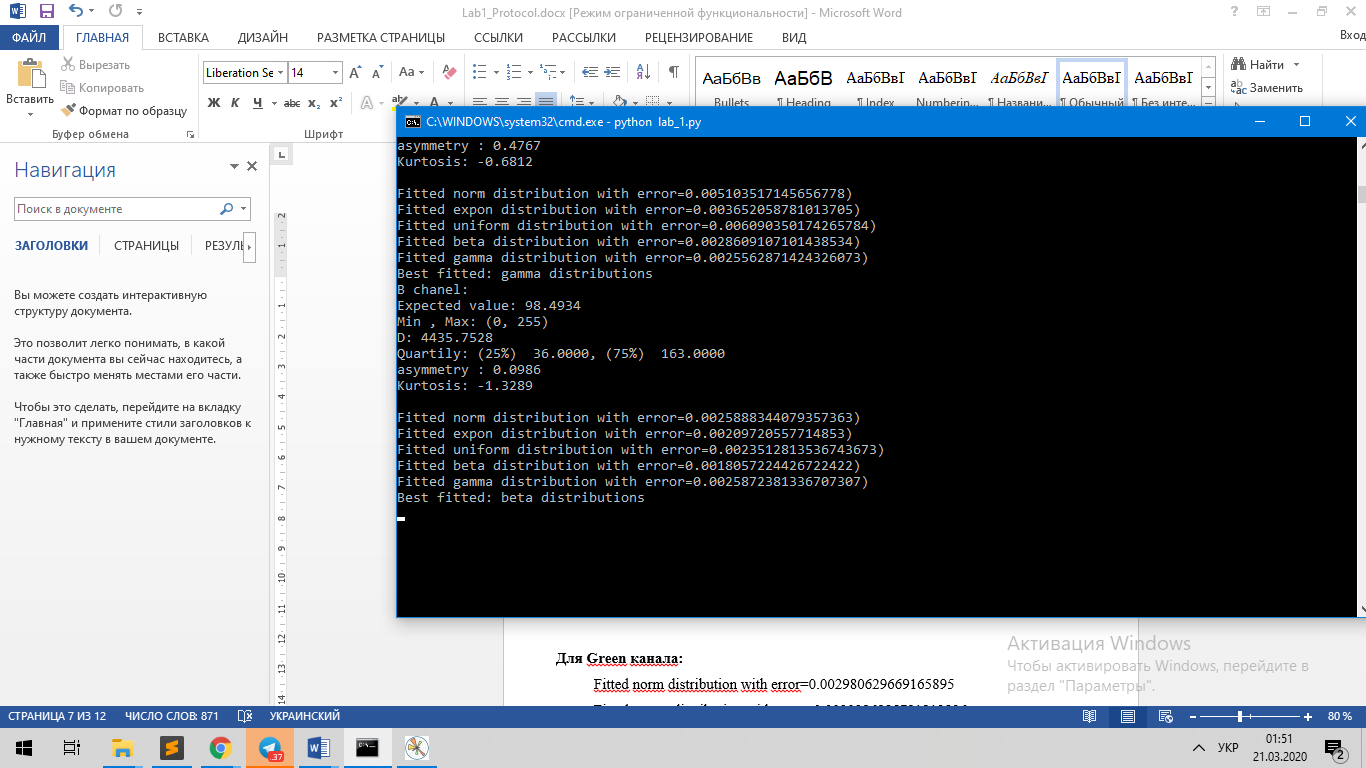
**Для Red каналу:**



**Для Green каналу:**



**Для Blue каналу:**



**Код програми**

*import random*

*from pathlib import Path*

*import numpy as np*

*import scipy.stats as st*

*from PIL import Image, ImageDraw*

*from fitter import Fitter*

*from matplotlib.pyplot import figure, title, xlabel, show*

*DATASET\_FILE = 'images\_dataset.txt'*

*DISTRIBUTIONS = ['norm', 'expon', 'uniform', 'beta', 'gamma']*

*CURRENT\_IMG\_DIR = Path(\_\_file\_\_).resolve() / 'current\_img'*

*def draw\_chanel(image, n):*

*width = image.size[0]*

*height = image.size[1]*

*pix = image.load()*

*draw = ImageDraw.Draw(image)*

*for i in range(width):*

*for j in range(height):*

*a, b, c = 0, 0, 0*

*if n == 0:*

*a = pix[i, j][0]*

*if n == 1:*

*b = pix[i, j][1]*

*if n == 2:*

*c = pix[i, j][2]*

*draw.point((i, j), (a, b, c))*

*image.save("current\_img/ans" + str(n) + ".jpg", "JPEG")*

*del draw*

*def get\_rgb\_channels(image):*

*width = image.size[0]*

*height = image.size[1]*

*pix = image.load()*

*RGB = [[], [], []]*

*for i in range(width):*

*for j in range(height):*

*a = pix[i, j][0]*

*b = pix[i, j][1]*

*c = pix[i, j][2]*

*RGB[0].append(a)*

*RGB[1].append(b)*

*RGB[2].append(c)*

*image.save("current\_img/ans.jpg", "JPEG")*

*return RGB*

*def get\_statistic\_info(x):*

*result = [np.mean(x), (np.min(x), np.max(x)), np.var(x), (np.percentile(x, 25), np.percentile(x, 75)),*

*st.mode(x), st.skew(x), st.kurtosis(x)]*

*return result*

*def main():*

*with open(DATASET\_FILE, 'r') as file:*

*image\_names = ['mirflickr/im' + x.strip() + '.jpg' for x in file.readlines()]*

*random.shuffle(image\_names)*

*for name in range(len(image\_names)):*

*print(image\_names[name])*

*rgb\_channels = get\_rgb\_channels(Image.open(image\_names[name]))*

*for channel in range(len(rgb\_channels)):*

*m, minmax, s, perct, mode, skew, kurt = get\_statistic\_info(rgb\_channels[channel])*

*channels = ['R', 'G', 'B']*

*print('{} chanel:'.format(channels[channel]))*

*print('Expected value: {0:.4f}'.format(m))*

*print('Min , Max: ({0}, {1})'.format(\*minmax))*

*print('D: {0:.4f}'.format(s))*

*print('Quartily: (25%) {0:.4f}, (75%) {1:.4f}'.format(\*perct))*

*print('asymmetry : {0:.4f}'.format(skew))*

*print('Kurtosis: {0:.4f}'.format(kurt))*

*print()*

*bins\_number = 256*

*figure()*

*res = Fitter(rgb\_channels[channel], distributions=DISTRIBUTIONS, bins=bins\_number)*

*res.fit()*

*fit\_result = res.\_fitted\_errors*

*key\_min = min(fit\_result.keys(), key=(lambda k: fit\_result[k]))*

*print(f'Best fitted: {key\_min} distributions')*

*res.summary()*

*res.hist()*

*title(f'Histogram of pixel brightness for {channels[channel]} channel for {image\_names[name]}')*

*xlabel('bright')*

*show()*

*draw\_chanel(Image.open(image\_names[name]), channel)*

*if \_\_name\_\_ == '\_\_main\_\_':*

*main()*

**ВИСНОВКИ**

У лабораторній роботі було розраховано значення статистичних характеристик зображень для кожного каналу кольорів, було побудовано гістограми яскравостей пікселів кольорових зображень по трьом каналам та отримані апроксимації цих гістограм відомими функціями розподілу випадкових величин.

Було показано похибку апроксимації для зображення для кожного каналу кольорів.

З отриманих гістограм на рисунках 1-3 можна зробити висновок, що найкраще апроксимує гістограми яскравостей пікселей це Beta-розподіл.