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»

Варіант №7

Керівник: Виконав:

Прогонов Дмитро Олександрович студент 5 курсу групи ФЕ-91мп

Захищено з оцінкою Нікішин М.Ю.

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

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

дата, підпис

Київ – 2020 р.

Завдання на лабораторну роботу:

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.

**Хід роботи**

**1) Формування тестової вибірки зображеннь**

Лабораторна робота буде виконана мовою Python за допомогою блокового інтерпритатора Jupyter. В данній лабораторній роботі використовувались наступні бібліотеки:

* Scipy
* Matplotlib.pyplot
* Os
* Numpy
* Seaborn
* Random
* Time
* Np
* PIL
* Pandas

Задаємо таблицю з необхідними нам параметрами:

* path = '.../downloads/mirflickr/im'
* folder = os.path.join(os.getcwd(), 'mirflickr/')
* images\_sample = []
* **for** i **in** random.sample(range(6,25001),250):
* images\_sample.append(path + str(i) + '.jpg')
* columns = ['num',
* 'channel',
* 'max',
* 'min',
* 'matspod',
* 'disp',
* 'median',
* 'quart',
* 'skew',
* 'kurt',
* 'data']

**2) Обчислення характеристик для кожного каналу зображень**

**def** do\_open(file):

im = Image.open(file)

**return** np.array(im)

**def** get\_sample():

t = time.time()

np\_arrays = map(do\_open, images\_sample)

array = [i **for** i **in** np\_arrays]

**return** array

**def** get\_df(image):

values = []

**for** i **in** range(len(image)):

**for** channel **in** range(3):

channel\_img = image[i][:,:,channel]

values.append(list((random.sample(range(6,25001),250)[i], channel,

channel\_img.max(),channel\_img.min(), np.mean(channel\_img),np.var(channel\_img),

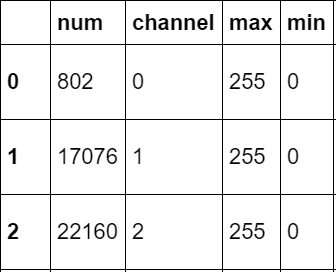
np.median(channel\_img), iqr(channel\_img), skew(channel\_img, **None**), kurtosis(channel\_img, **None**),

channel\_img)))

**return** pd.DataFrame(values, columns=columns)

1. Максимальна / мінімальне значення

отимаємо наступні значення для першого зобрадення(де канал 0 це червоний колір; канал1 – блакитний; канал 2 – зелений):



RED:

Max:255, Min: 0

BLUE:

Max:255, Min: 0

GREEN:

Max:255, Min: 0

1. Математичне сподівання і дисперсія

Розрахунки будуть проводитись за наступними формулами:



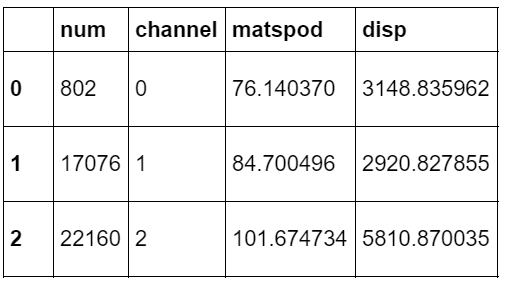
(1),



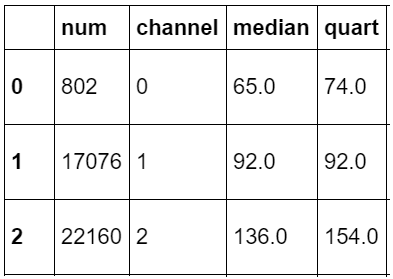
(2),

Де (1) - математичне очікування, а (2) – дисперсія, xi - значення яскравості, pi – ймовірність її появи. pi можна знайти як кількість пікселів даної яскравості поділену на всю кількість пікселів.

Таким чином отримуэмо значення Мат. очікування та дисперсії:



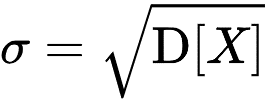
с. Медіана значень та інтерквартальний розмах.

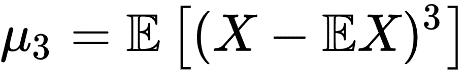


d. Коефіцієнти асиметрії та ексцесу

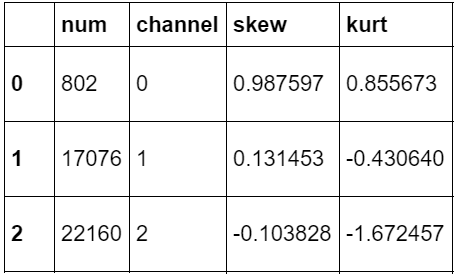
Використаємо відповідні формули коефіцієнту асиметрії:







Отримаємо:



1. **Для кожного каналу кольору кожного зображення з тестового пакета побудувати гістограму значень яскравості пікселів**

Звернемося до раніше використанних бібліотек,

**def** get\_hist(m, n):

plt.figure(figsize=(12,9))

**if** n == 0:

print('red')

**elif** n == 1:

print('green')

**else**:

print('blue')

plt.subplot(m),plt.hist(df.loc[df['channel'] == n]['data'])

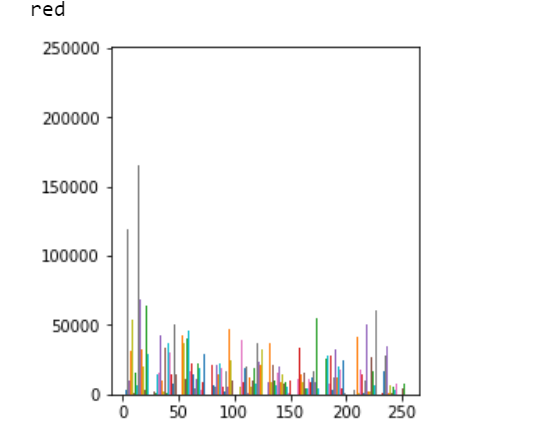
plt.show()

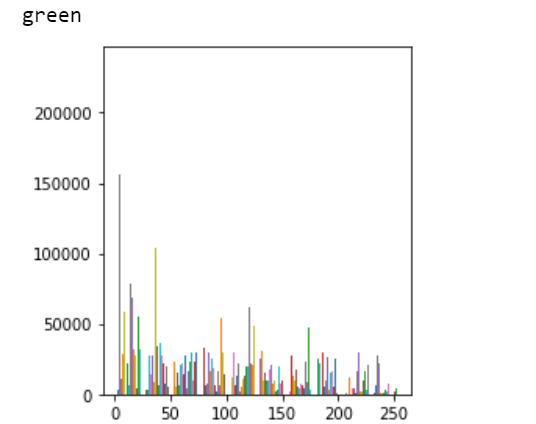
**return** X, Y

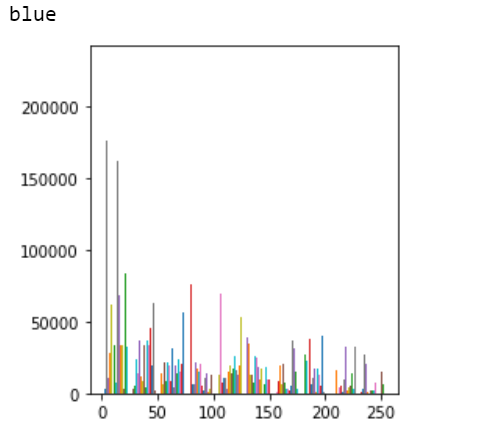
r\_hist = get\_hist(231, 0)

g\_hist = get\_hist(232, 1)

b\_hist = get\_hist(233, 2)

****

****

****

1. **Провести апроксимацію отриманих гістограм з використанням відомих імовірнісних розподілів, визначити найкращу апроксимацію**

Для даного завдання звернемося до раніше використанної бібліотеки

**def** get\_approximate():

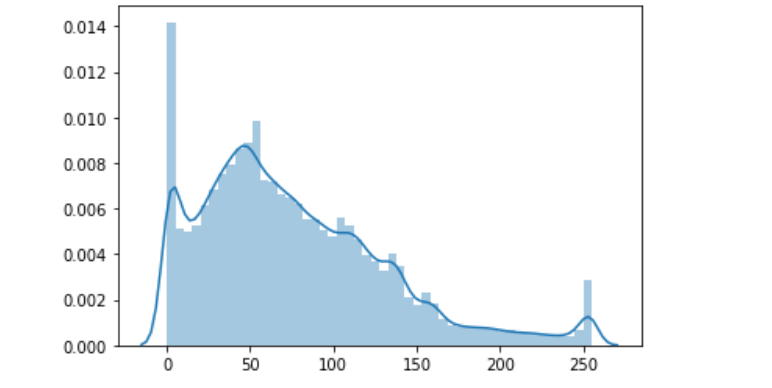
hist\_data = df.loc[df['channel'] == 0]['data']

test = hist\_data[0]

X,Y = sns.distplot(np.ravel(test)).get\_lines()[0].get\_data()

**return** test, X, Y

test, X, Y = get\_approximate()



Побудована апроксимація описує наші дані, тепер використаемо відомі розподіли:

**def** get\_best(test):

test = np.ravel(test)

x\_min, \_max = min(plt.xticks()[0] ), max(plt.xticks()[0] )

m, s = stats.norm.fit((X,Y))

do\_norm = stats.norm.pdf(np.linspace(0, 255, len(X)), m, s)

plt.plot(np.linspace(0, 255, len(X)), do\_norm, label='Norm',color = 'grey')

plt.plot(X,Y,color = 'black',linewidth = 4)

plt.plot(np.linspace(0, 255, len(X)), do\_norm, label='Norm')

ag,bg,cg = stats.gamma.fit((X,Y))

do\_gamma = stats.gamma.pdf(np.linspace(0, 255, len(X)), ag, bg,cg)

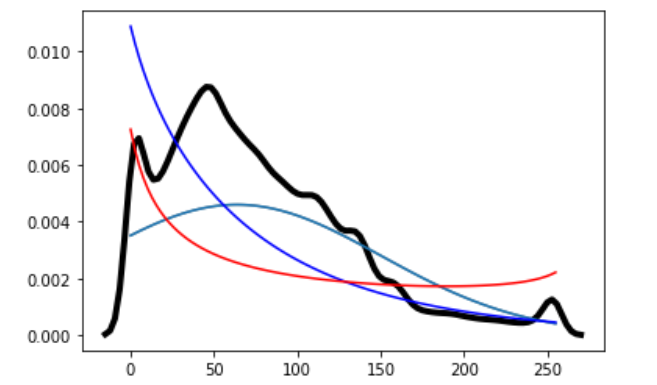
plt.plot(np.linspace(0, 255, len(X)), do\_gamma, label='Gamma',color = 'blue')

ab,bb,cb,db = stats.beta.fit((X,Y))

do\_betta = stats.beta.pdf(np.linspace(0, 255, len(X)), ab, bb,cb, db)

plt.plot(np.linspace(0, 255, len(X)), do\_betta, label='Beta',color = 'red')

plt.show()



Таким чином, до нашого розподілу апроксимує Beta.

**ВИСНОВКИ**

У данній лабораторній роботі було розраховано значення статистичних характеристик вибірки 250 зображення для кожного каналу кольорів. Було знайдено що всі канали охоплюють увесь спектр значень.

Розраховано мат.очікування для кожного кольору:

RED: matspod - 76.140370; disp - 3148.835962

BLUE: matspod - 84.700496; disp - 2920.82785

GREEN: matspod - 101.674734; disp - 5810.870035

Медіану значень та інтерквартальний розподіл:

RED: median - 65.0; quart - 74.0

BLUE: median - 92.0; quart - 92.0

GREEN: median - 136.0; quart - 154.0

Також було знайдено коефіцієнти асиметрії та коефіцієнт ексцесу:

RED: skew - 0.987597; kurt - 0.855673

BLUE: skew - 0.131450; kurt - -0.430640

GREEN: skew - -0.103828; kurt - -1.67245

Так, як коефіцієнт асиметрії додатній, то напрямок грайіка буде в бік спадання значень.

Графіки приведено за допомогою бібліотеки matplotlib.pylp, вони повністю відповідають знайденим значенням.

При знаходженні апроксимації було показано топ відомих розподілів по схожості з нишим. Було отримано що бета-розподіл підходить краще інших.