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 № 3

з дисципліни: «Автоматизація обробки ІзОД»

Варіант №9

Керівник: Виконав:

Прогонов Дмитро Олександрович студент 5 курсу групи ФЕ-91мп

Захищено з оцінкою Проценко Дмитро Олександрович

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

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

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. Математичне сподівання і дисперсію;
   2. Коефіцієнти асиметрії та ексцесу (нормалізований);
3. Використовуючи моделі SPAM і CC-PEV (<http://dde.binghamton.edu/download/feature_extractors/>), розрахувати вектора характеристик кожного каналу кольору кожного зображення;
4. Отримані параметри зображень упакувати в матрицю ознак (кожен рядок відповідає результатам для окремого каналу кольору тестового зображення, кожен стовпець - параметру зображення). Матриці ознак побудувати окремо для статистичних характеристик зображень (1-4 центральні моменти), а також моделей SPAM и CC-PEV;
5. Побудувати вектор міток класів зображень. Назва цільового класу для кожного студента визначається згідно з позицією студента у списку групи (див. Вкладений файл - наприклад, для першого студента в списку групи цільової клас "explore", для другого студента - мітка "sky", для третього студента - мітка "nikon" і т.д.);
6. псевдовипадковий чином розділити вихідний пакет зображень на 2 рівні частини (тестова і контрольна підвибірки). З використанням тестової підвибірки провести настройку наступних класифікаторів:
   1. Лінійна регресія;
   2. Робастна регресія;
   3. Логістична регресія;
   4. Метод опорных векторів (SVM);
7. Використовуючи налаштовані класифікатори з п. 6 провести обробки зображень з контрольного підпакету. Оцінити ймовірності правильної класифікації (0 і 1 класи), а також ймовірності помилок першого (помилкове спрацьовування) і другого (пропуск цілі) роду;

Повторити пп. 6-10 разів для отримання усередненої точності класифікації

**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
* Os
* Numpy
* Pandas
* Sklearn

Тестова вибірка сформована із псевдовипадкового вибору 250 кольорових зображень із пакету mirflick.

Приклад коду для вибірки зображень:

*# load all images in a directory*

random.seed(9)

random\_indexes = random.sample(range(25000), 250)

loaded\_images = list()

**for** i **in** range(250):

filename = 'im' + str(random\_indexes[i]) + '.jpg'

img\_data = image.imread('mirflickr25k/mirflickr/' + filename)

loaded\_images.append(img\_data)

print('> loaded **%s** **%s**' % (filename, img\_data.shape))

Тепер отриманий масив зображень буде знаходитись в loaded\_images в виді двомірного масиву з трьома значеннями яскравості в кожрній комірці.

Далі сформуємо матрицю для збору статистичних даних про зобьраження. Для цього створимо двомірний Numpy масив на три рядки для кожного каналу кольору та на 256 стовпчиків, що відповідатиме кількості пікселів відповідної яскравості.

values = np.zeros((3, 256))

index = 0

**for** image **in** loaded\_images:

**for** i **in** range(image.shape[0]):

**for** j **in** range(image.shape[1]):

values[0][image[i][j][0]] += 1

values[1][image[i][j][1]] += 1

values[2][image[i][j][2]] += 1

index += 1

**if** (index % 10) == 0:

print('> processed **%i** files' % (index))

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

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

*#GREEN*

sum\_value = sum(values[GREEN])

M\_green = 0

**for** index **in** range(len(values[GREEN])):

p = (values[GREEN][index] / sum\_value)

M\_green += p \* index

D\_green = 0

**for** index **in** range(len(values[GREEN])):

p = (values[GREEN][index] / sum\_value)

D\_green += p \* ((index - M\_green) \*\* 2)

print("Green: expected value (math. ochikuvanya) : **{0:.2f}**, dispersion : **{1:.2f}**"\

.format(M\_green, D\_green))

Таким чином отримуэмо значення Мат. очікування та дисперсії:

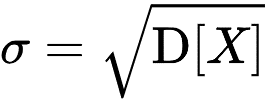
Red: expected value (math. ochikuvanya) : 109.15, dispersion : 6344.77

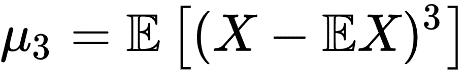
Green: expected value (math. ochikuvanya) : 101.70, dispersion : 5744.94

Blue: expected value (math. ochikuvanya) : 93.15, dispersion : 6225.48

1. Коефіцієнти асиметрії та ексцесу

Використаємо відповідні формули коефіцієнту асиметрії:





**def** E\_operator(arr\_values, M, power):

sum\_val = sum(arr\_values)

ans = 0

**for** index **in** range(len(arr\_values)):

p = (arr\_values[index] / sum\_val)

ans += p \* ((index - M) \*\* power)

**return** ans

*#asymmetry, ekscess*

Asym\_red = E\_operator(values[RED], M\_red, 3) / (D\_red \*\* (3 / 2))

Asym\_green = E\_operator(values[GREEN], M\_green, 3) / (D\_green \*\* (3 / 2))

Asym\_blue = E\_operator(values[BLUE], M\_blue, 3) / (D\_blue \*\* (3 / 2))

Ekscess\_red = E\_operator(values[RED], M\_red, 4) / (D\_red \*\* 2)

Ekscess\_green = E\_operator(values[GREEN], M\_green, 4) / (D\_green \*\* 2)

Ekscess\_blue = E\_operator(values[BLUE], M\_blue, 4) / (D\_blue \*\* 2)

print('Red: asymmetry - **{0:.3f}**, ekscess - **{1:.3f}**'\

.format(Asym\_red, Ekscess\_red))

print('Green: asymmetry - **{0:.3f}**, ekscess - **{1:.3f}**'\

.format(Asym\_green, Ekscess\_green))

print('Blue: asymmetry - **{0:.3f}**, ekscess - **{1:.3f}**'\

.format(Asym\_blue, Ekscess\_blue))

Отримаємо:

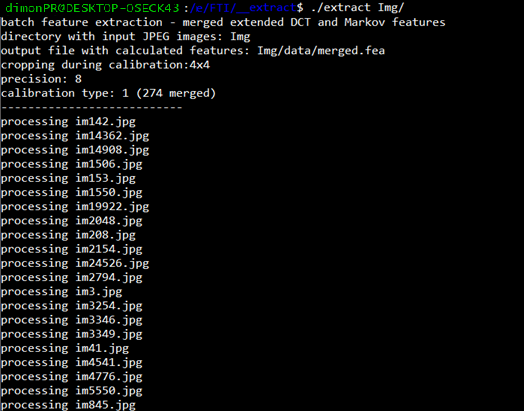
Red: asymmetry - 0.283, ekscess - -1.155

Green: asymmetry - 0.366, ekscess - -1.028

Blue: asymmetry - 0.544, ekscess - -0.956

**3 - 4. Знаходження векторів ознак за допомогою функції CC-PEV**

Використаємо екстрактор CC-PEV для знаходження векторів ознак зображень:



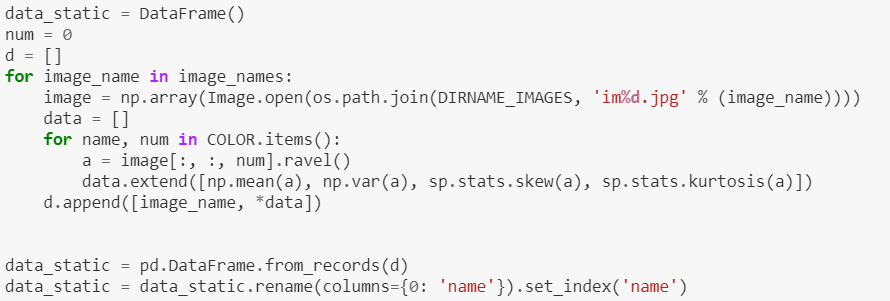
Після виконання пограми буде сформовано вектор ознак для кожного фото та поміщено в файл merged.fea.

**5. Побудувати вектор міток класів зображень**

Тепер сформуємо вектори з мітками мого варіанту. В даному випадку це «portrait». Теги отримуємо з папки tags пакету.



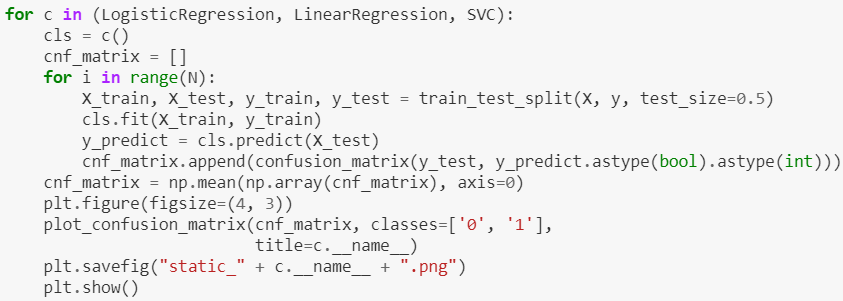
Тепер сформуємо пакети даних та міток для навчання:

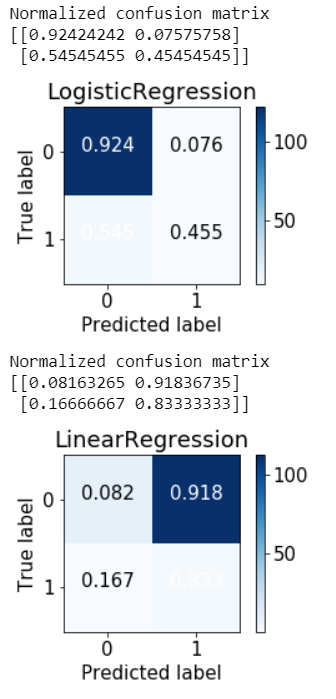


1. **Навчання моделі на першій вибірці**



Лінійна та Логістична регресія:





**ВИСНОВКИ**

В даній лабораторній роботі було проаналізовано вибірку з 250 зображень датасету mirflickr-20k. Було знайдено що всі канали охоплюють увесь спектр значень. Було знайдено мат. очікування – 109.15 для чевоного каналу, 101.70 для зеленого і 93.15 для синьогоканалу відповідно і дисперсію 6344.77, 5744.94 та 6225.48. Також знайдено коефіцієнти асиметрії та ексцесу:

Червоний: асиметрія - 0.283, ексцес -1.155

Зелений: асиметрія - 0.366, ексцес -1.028

Синій: асиметрія - 0.544, ексцес - 0.956

Було проведено формування ознак на основі CC-PEV екстрактора та сформовано вектори данних для подальшого аналізу. Також всі дані було упаковано в матрицю ознак для тренування моделей.

Використовуючи мітку цільового класу «portrait» було сформовано дані для розрізняння що відповідають та не відповідають моделі.

Післф цього було натреновано моделі Лінійної та Логістичної Регресії для розрізняння. За даними таблиць результатів бачимо що він складав 0.918 та 0.924 відповідно для двох моделей.