

Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Лабораторна робота №3

Виконали:
студенти КА-77
Буркацький Микита
Морозов Роман
Котів Сергій

Завдання

а) Для початку реалізувати **енкодер та декодер** (плеєр) для відеоряду на основі методів оцінки руху. Відповідно вхідним файлом буде відеозапис (наприклад трейлер до Тенет), одним з виходів буде файлик у вашому форматі що містить гіпотетично стиснений відеоряд з наприклад інформації щодо руху кожного третього кадру і два сусідні у повному розмірі (MPEG-7 алгоритми з що ми їх дотично згадували попередніх лекціях будуть в нагоді). Другим виходом буде власне відтворення відео з вашого файлу. Після того як ви налаштували ваш енкодер і плеєр, йдете до друга по команді що в цей час налаштовував свої енкодер і плеєр на основі іншого алгоритму апроксимації руху і ви починаєте об'єднувати зусилля: користуючись вашими енкодерами ви генеруєте ваші апроксимовані зображення та обчислюєте їх різницю в порівнянні з дійсним зображеннями з оригінального відеоряду, збираючи при цьому візуальні різниці для кожного з методів. Окрім цього слід зібрати метрику швидкодії енкодерів і плеєрів, порівняти отримані результати і заліти їх аналогічним чином в гуглдок. Бонусні бали за опцію datamoshing і запис та відтворення аудіо, підтримку більш ніж одного формату відео, сприйняття стрімів з Youtube.

б) Маючи досягнення з попередньої серії, докрутити до лаби **класифікатор** на ваш особистий смак, натренувати його на ваш об'єкт (або ж якщо є час і натхнення то на інший), за необхідності розширюючи вибірку, і на об'єкт сусіда з таким же класифікатором на основі алгоритмів отримання характеристик зображення. Після того зробити дві речі: прогнати отримане комбо з класифікатора і дескриптора на тестовій вибірці збираючи як позитивні результати так і помилки першого і другого роду та замірюючи швидкодію; і записати демонстраційне відео з вашим об'єктом та згодувати його цьому ж комбо записуючи у окремий файл результат роботи (скажімо те ж відео з текстовим оверлеєм) і також обчислити кількість помилок першого та другого роду. Далі знаючи особливості вашого дескриптора колективно з кимсь у кого такий же класифікатор проаналізувати особливості класифікатора в комбінації з вашими дескрипторами. Бонусні очки за використання кількох класифікаторів (видів класифікаторів) і/або запис відео з візуалізацією процесу розпізнавання/детекції.

Обмеження таке ж як в попередній лабі, пари алгоритмів мають бути унікальними в кожній команді. Команди в тому ж файлику, що й на попередню лабу, в іншій вкладці.

Назва команди

Наша команда називається «Комп'ютерний зір».

Дескриптори

SIFT (Scale-Invariant Feature Transform)

ORB (Oriented FAST and Rotated BRIEF)

BRIEF (Binary Robust Independent Elementary Features)

Предмети

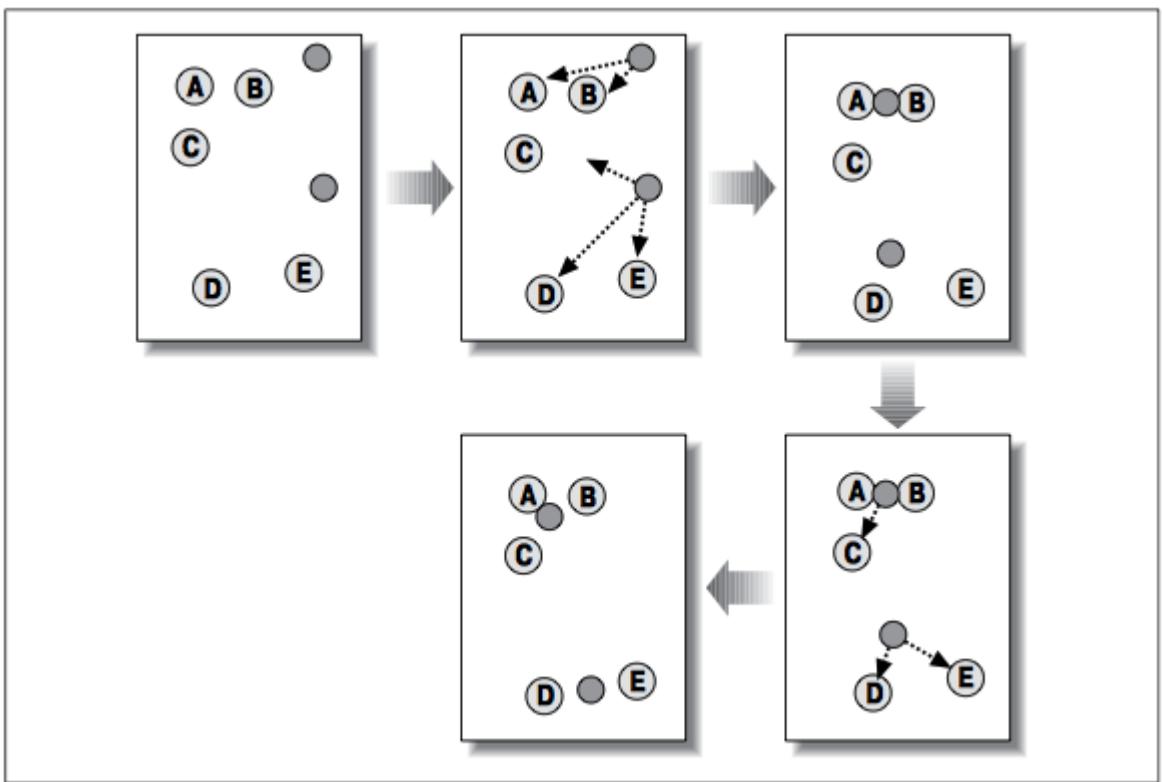
Книжка1, сірники, книжка2.

Ми обрали другий варіант виконання цієї лабораторної, алгоритм якого можна представити у вигляді: дескриптор – кластеризація – bag-of-words – класифікатор.

Теоретична частина

Дескриптором ключової точки називається числовий вектор, що характеризує особливості зображення в околиці даної точки. Спочатку процедура побудови дескрипторів застосовувалася для знаходження відповідності між ключовими точками на різних зображеннях. В результаті побудови формується безліч векторів ознак для вихідного набору особливих точок. В основі SIFT лежить ідея обчислення гістограми орієнтованих градієнтів в околиці особливої точки. Мета створення дескриптора BRIEF полягає в тому, щоб забезпечити розпізнавання однакових ділянок зображення, які були отримані з різних кутів огляду. При цьому ставиться завдання зменшити кількість виконуваних обчислень. В околиці точки вибирається деяким чином безліч пар пікселів і на даному безлічі будується набір бінарних тестів, які представляють собою порівняння інтенсивності у відповідних пікселях. Результати тестів об'єднуються в бітовий рядок, яка і є підсумковим дескриптором ключовий точки. Більш ефективною альтернативою дескриптору BRIEF є бінарний дескриптор ORB.

Кластеризація даних - це метод виявлення і візуалізації груп пов'язаних між собою предметів. Використовується в додатках, що обробляють великі обсяги даних. Кластеризація - приклад навчання без вчителя. На відміну від нейронних мереж або дерев рішень, алгоритмам навчання без вчителя не повідомляються правильні відповіді. Їх завдання - виявити структуру в наборі даних, коли жоден елемент даних не є відповідю. Кластеризація методом K-середніх починається з вибору k випадково розташованих центройд (точок, що представляють центр кластера). Кожному елементу призначається найближчий центр ваги. Після того як призначення виконано, кожен центр ваги переміщується в точку, яка розраховується як середнє по всім приписаним до нього елементів. Потім призначення виконується знову. Ця процедура повторюється до тих пір, поки призначення не припинять змінюватися.



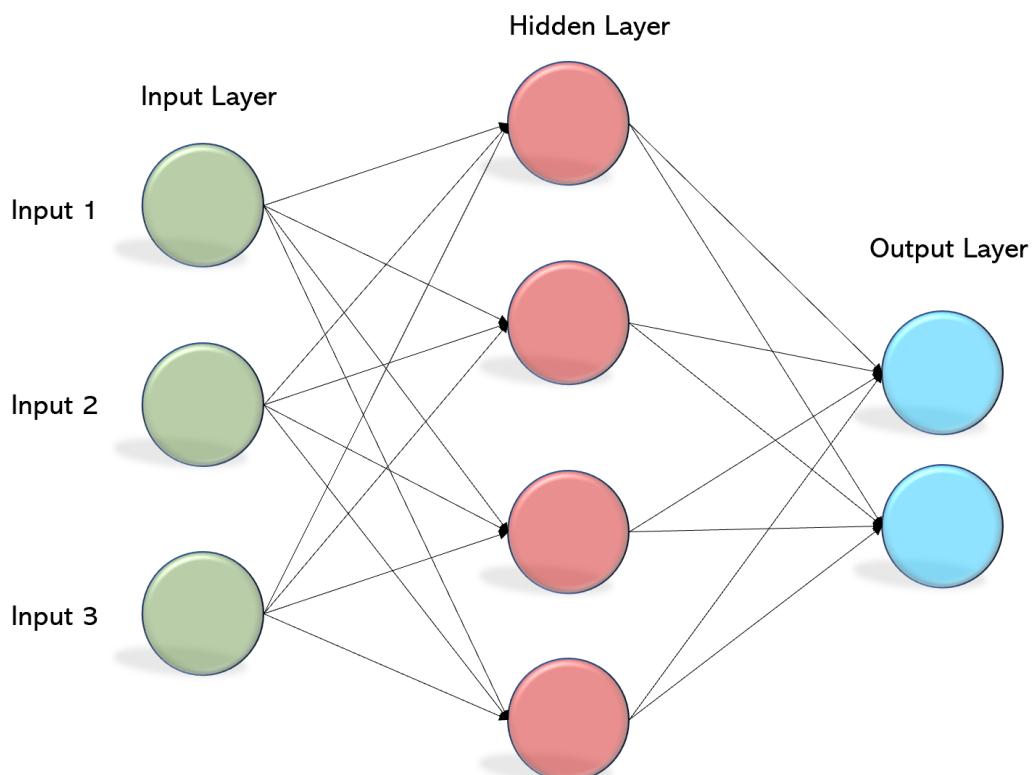
На першому кроці випадково розміщені два центроїда (зображені темними кружками). На другому кроці кожен елемент приписаний до найближчого центроїда. В даному випадку А і В приписані до верхнього центроїда, а С, Д і Е - до нижнього. На третьому кроці кожен центр ваги переміщений в точку, розраховану як середнє по всім приписаним до нього елементів. Після перерахунку призначень виявилося, що С тепер ближче до верхнього центроїда, а Д і Е - як і раніше близьче до нижнього. Остаточний результат досягається, коли А, В і С поміщені в один кластер, а Д і Е - в інший. В даному методі, дуже важливий вибір центроїдів, тому що в кінцевому підсумку все точки до них стягнуться, але якщо все 4 будуть дуже близькі кольору і з них будуть складатися великі області, то в кінцевому підсумку ви і отримаєте приблизно однакові центри. Таким чином центроїди повинні бути максимально рознесені (рівномірно і рівновіддалено).

Більшість відомих алгоритмів навчання з учителем вимагає опису кожного об'єкта у вигляді векторів однакової довжини. У зв'язку з цим безпосередньо використовувати обчислені значення дескрипторів, об'єднані в один вектор, для вирішення задачі класифікації не представляється можливим, тому що в загальному випадку число ключових точок на зображеннях різне. До того ж, таке уявлення в будь-якому разі не характеризувалося би інваріантністю щодо багатьох перетворень зображення, тому що при цьому зазвичай змінюється розташування даних точок і, отже, розташування дескрипторів всередині об'єднаного вектора. Данна проблема вирішується при використанні bag-of-words підходу, в якому зображення описується не самими дескрипторами, а частотами зустрічей їх різних типів на зображені. Типи дескрипторів описуються словником, який містить центроїди кластерів

дескрипторів особливих точок. Таким чином, в разі, коли словник уже обчислений, опис зображення обчислюється за наступною схемою:

- детектування ключових точок на зображенні і обчислення їх дескрипторів
- знаходження для кожного обчисленого дескриптора найближчого до нього центроїда кластера
- обчислення опису зображення у вигляді нормованої гістограми (i-ий бін гістограми відповідає числу входжень i-го слова зі словника в опис зображення, що відповідає числу дескрипторів, віднесенних до i-го кластеру)

Багатошаровий персепtron - це клас штучних нейронних мереж прямого поширення, що складається як мінімум з трьох шарів: вхідного, прихованого і вихідного. За винятком вхідних, все нейрони використовують нелінійну функцію активації. При навчанні MLP використовується навчання з учителем і алгоритм зворотного поширення помилки.



MLP показав можливість знаходити наближені рішення для надзвичайно складних завдань. Зокрема, вони є універсальним апроксиматором функцій, тому з успіхом використовуються в побудові регресійних моделей. Оскільки класифікацію можна розглядати як окремий випадок регресії, коли вихідна змінна категоріальна, на основі MLP можна будувати класифікатори.

Висновки

Для навчання ми використали 263 фото з 4 класів - сірникова коробка, книжка чорна, книжка кольорова, та відсутність будь-якого з цих предметів

для тестування - 120 фото, по 30 фото з кожного класу. Класифікатор – багатошаровий персептрон.

```
### ANOTHER NEURAL NETWORK
model = Sequential([
    Dense(20, activation = 'relu'),
    Dense(3, activation='softmax')
])
```

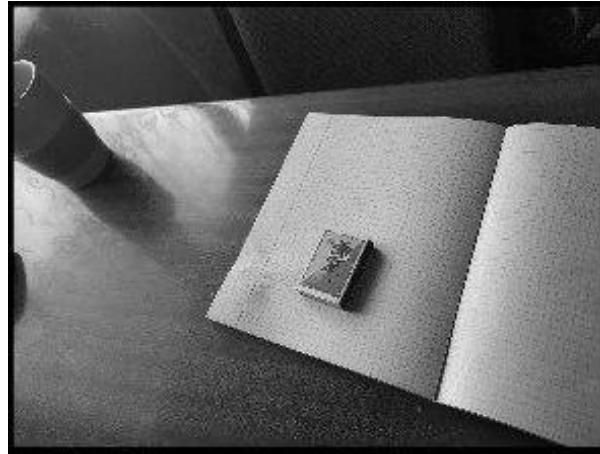
SIFT

Порівняно низька якість розпізнавання пов'язана із великою кількістю ключових точок із високою похибкою локалізації. Якби ми відсіяли певну частину цих точок, то точність би напевне виросла.

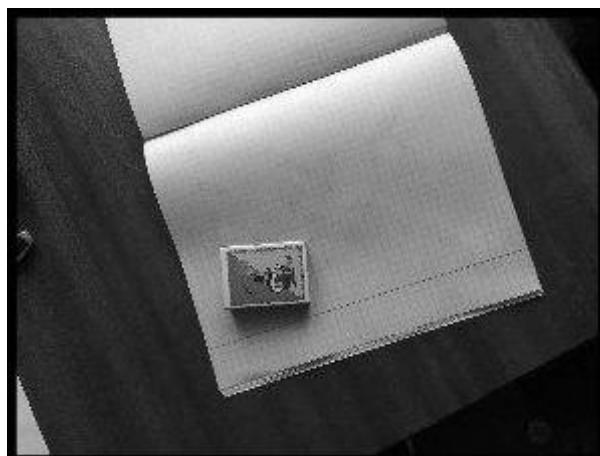
Точність (метрика accuracy в tensorflow) - 69.5
Помилки
Сірники - 13/30
Книга чорна - 11/30
Книга біла - 8/30
Немає об'єкту - 20/30

Чому так сталося?

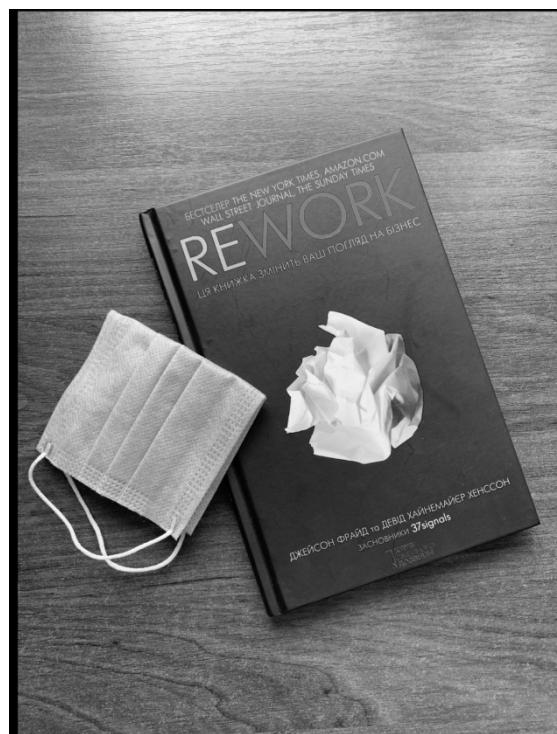
- На сірниках найвища помилка першого роду, через високу різноманітність датасету. Це означає те, що в нас є по декілька фотографій сірникової коробки в різних ситуаціях, чого, очевидно, недостатньо для високоточної роботи класифікатора-персептрона. Більш того, для кращої працездатності процесора, перед обчисленням ключових точок та складання словника bag-of-words, всі фото були змінені в розмірі методами opencv - зменшені в 4 рази. Це зменшує кількість ознак, що ми можемо витягнути із такого датасету.
- Датасет чорної книги має в собі найменшу різноманітність. Але у попередній роботі ми переконалися, що на чорній книзі знаходиться мала кількість ключових точок, тому що основна частина книги - чорний фон. Тому і результати трохи кращі, ніж на сірниках. Тестовий сет був доволі схожий на навчальний.
- Кольорова книга має трохи різноманітніший датасет, ніж чорна, а також на ній зазвичай більше ключових точок. Тестовий сет був схожий на навчальний, а також через кількість ключових точок з дескрипторами ми отримали відносно непогану точність.
- Зрозуміло, що щоб отримувати хороші результати на фото, де немає об'єкту потрібно дуууже великий датасет із різними обстановками, тому помилка другого роду така висока.



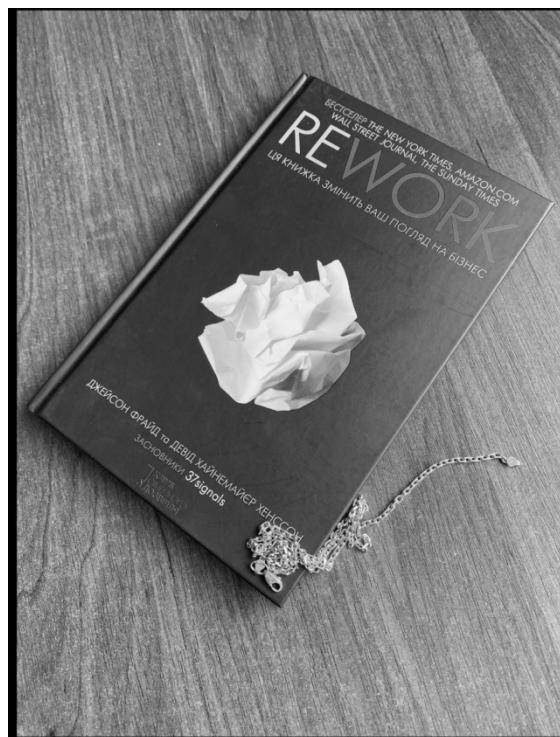
Типовий приклад помилки SIFT+MLP. Доволі великий поворот тестового зображення, на яке погано реагує дескриптор.



В цьому і попередньому прикладі алгоритм сплутав коробку із кольоровою книгою - можливо через білий зошит на якому лежить коробка.



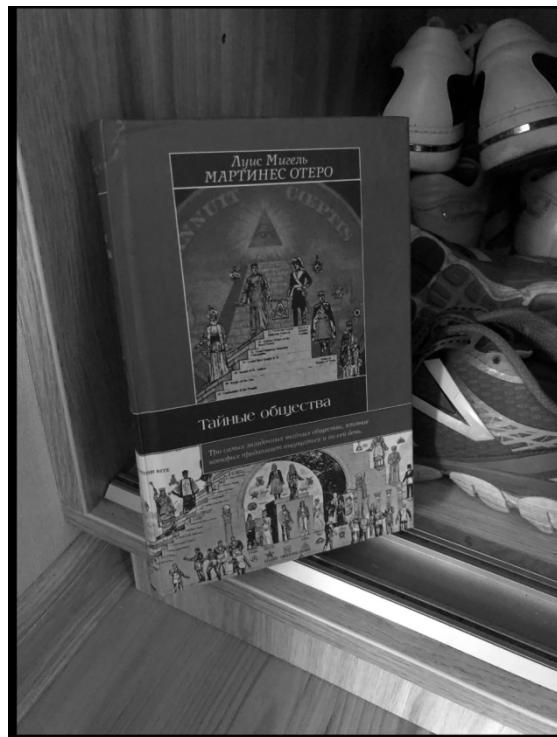
Цікавий приклад - алгоритм вказав на те, що тут немає об'єкта.



Сплютали чорну книгу з кольоровою, хоча поворот невеликий, і фото подібне на фото із навчальної вибірки.



Сплютали кольорову книгу із чорною - одноманітний фон, хороше світло.



Алгоритм вказав, що на фото немає предмету, можливо через доволі велику кількість ключових точок поза книгою.

ORB

При роботі алгоритму ми знову зменшували фото в 4 рази, тепер вже для "чесного" порівняння із SIFT.

Точність (метрика в tensorflow) - 83.5

Помилки

Сірники - 4/30

Чорна книжка - 2/30

Кольорова книжка - 7/30

Немає предмету - 25/30

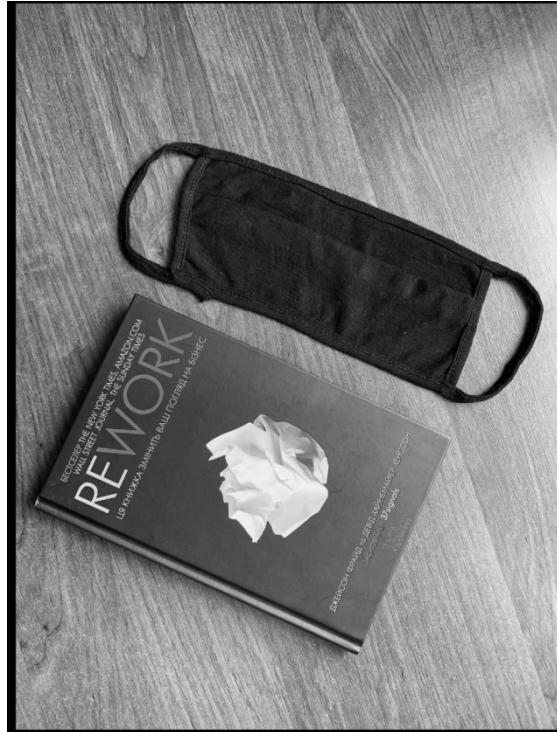
Порівняно із SIFT, алгоритм працює точніше. Причиною тому є менша кількість ключових точок, що визначає алгоритм, і краща їх якість. Помилки першого роду знизились у всіх предметів, це означає що алгоритм більш явно визначав ті риси, що притаманні предметам на фото. Водночас, помилка другого роду збільшилась. Скоріше всього це сталося через те, що довжина одного дескриптора в ORB - 32, проти 128 у SIFT, тому один дескриптор містить менше інформації про ключову точку, а отже і буде вказувати на схожість пустоти до якогось предмету із більшою ймовірністю. Другою причиною є розмір датасету, що не містить достатнього різноманіття сцен, а отже і дає помилки.



Алгоритм вказав на це розмите фото коробки як на чорну книжку.



Тут алгоритм сказав, що предмету немає. Швидше за все, фото цих сцен були відсутні в навчальній вибірці



Тут алгоритм сказав, що предмету немає. Мабуть, ключових точок було мало чи недостатньо описані.



Вже вдруге алгоритм називає це фото сірниковою коробкою.



Алгоритм сказав, що це фото - сірникова коробка. Багато помилок пов'язані із сірниковою коробкою - швидше за все через різноманітність дата сету.

BRIEF

Результати його роботи на одному рівні із SIFT. В попередній роботі ми визначили вже те, що цей алгоритм не знаходить багато ключових точок, і його дескриптори довжини 32, тому не містять так багато інформації, як SIFT, але в середньому його дескриптори кращої якості, знову тут впливає те, що ми не відсіювали ключові точки низької якості у інших алгоритмів.

Точність (метрика в tensorflow) - 67

Помилки

Сірники - 11/30

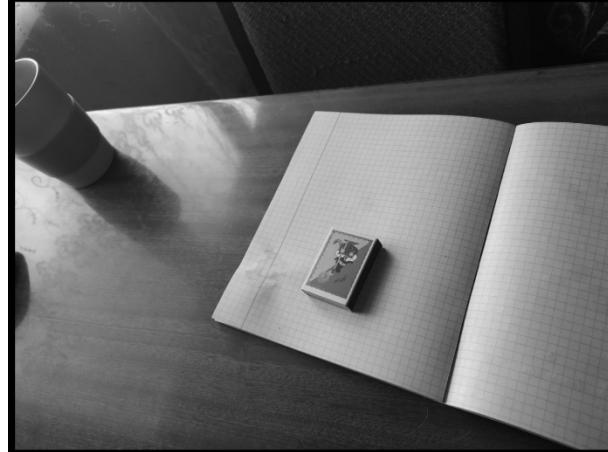
Чорна книжка - 13/30

Кольорова книжка - 5/30

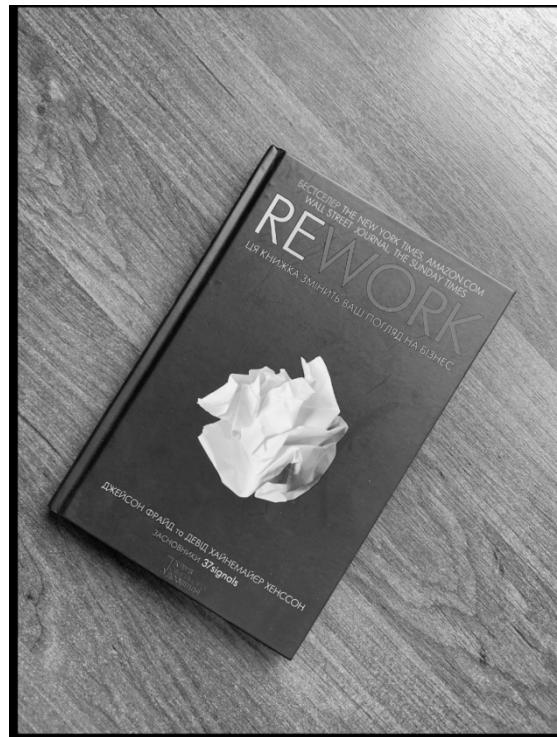
Немає предмету - 25/30



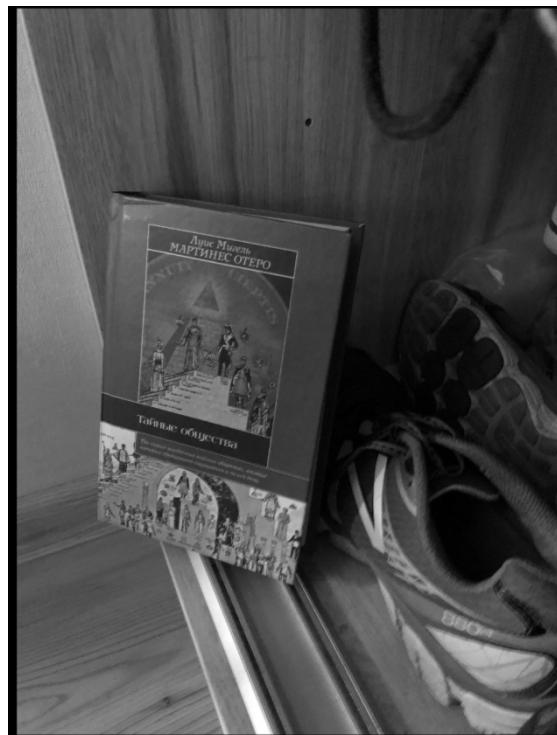
Сказано, що тут немає предмету. Швидше за все, це сталося через сильний поворот коробки.



Знову сказано, що тут кольорова книжка. Може бути через сильний поворот та інші предмети в кадрі з білим фоном.



Сказано, що на фото немає предмету. Схоже, мала кількість дескрипторів.



Сказано, що це чорна книжка.



Сказано, що це чорна книжка. Наймовірніше через фон, що використовувався у вибірці із тією книжкою.

Додаткове завдання:

Results

- <https://github.com/morozov-roman/CVPR>

References

- https://intuit.ru/studies/professional_skill_improvements/11339/courses/1106/lecture/18032?page=3
 - http://py-algorithm.blogspot.com/2014/10/blog-post_12.html
 - <https://wiki.loginom.ru/articles/multilayered-perceptron.html>
 - <https://stackabuse.com/overview-of-classification-methods-in-python-with-scikit-learn/>