

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

«До захисту допущено»

Завідувач кафедри ПМА

Чертов Олег Романович

«\_\_» \_\_\_\_\_ 2024

## Дипломна робота

на здобуття степеня бакалавра

за освітньо-професійною програмою «Наука про дані та математичне  
моделювання»

спеціальності 113 «Прикладна математика»

на тему: «Автоматичне маркування зображень за допомогою нейронних мереж»

Виконав:

студент IV курсу, групи КМ-01

Скорденко Д. О.

\_\_\_\_\_

Керівник:

доцент кафедри ПМА

Сирота С. В.

\_\_\_\_\_

Консультант з нормоконтролю:

доцент кафедри ПМА

Мальчиков В. В.

\_\_\_\_\_

## АНОТАЦІЯ

В даній роботі описано систему маркування зображень, в якій зроблено акцент на чотирьох аспектах: висока точність, використання додаткової модальності, явна підсистема для передбачення к-сті лейблів, простота компонентів системи.

**Актуальність теми:** Все частіше виникає задача маркування зображення. Деякі із цих задач: аналіз рентгенівських знімків, супутникових знімків, звичайних фотографій. Для досягнення високої точності важливо аналізувати всі можливі аспекти даних які надходять разом із задачею. Так для рентгенівських знімків - це може бути історія хвороб пацієнта, для супутникових знімків - різні метадані, геолокація тощо, а для аналізу звичайних фотографій - теги, анотації, метадані, тощо. Відносно новим застосуванням даної моделі може бути структуризація зображень, згенерованих ШІ, наприклад Stable Diffusion, де згенероване зображення - це основна модальність даних, а запит для його генерування (prompt) - це додаткова. Дана робота розглядає модульну систему, яку досить таки легко адаптувати під кожну із наведених вище задач та досягнути високої точності.

**Мета дослідження:** Метою даної роботи є розробка ПЗ для маркування зображень (шпалерів робочого столу) для покращення системи категоріального пошуку зображень (шпалерів робочого столу).

**Завдання дослідження:** Створити систему, яка буде виконувати маркування шпалерів на основі двох типів даних: зображення та шумні теги, які опціонально надаються користувачами при завантаженні зображення на хостинг.

Також необхідно розв'язати наступні задачі:

- Провести аналіз існуючих рішень
- Змодельовати систему
- Провести тренування системи
- Провести аналіз ефективності компонентів системи
- Порівняти якість і повноту опису розглянутої моделі відносно існуючих рішень на основі тестових метрик
- Навести ілюстративні приклади роботи системи

**Об'єкт дослідження:** Об'єктом дослідження є маркування зображень, та методи покращення маркування зображення. Для порівняння ефективності маркування серед існуючих рішень було обрано моделі, для яких обраховані тестові метрики для того ж датасету, який обрано в даній роботі. До множини існуючих рішень належать: SR-CNN-RNN [10], Resnet-SRN [22], MS-CMA [21], Query2Label [11], Resnet-CPSD [20] та ін.

**Предмет дослідження:** Предметом дослідження є множина шпалерів робочого столу в якості основної модальності даних, та додаткова інформація (надані людьми шумні теги) в якості додаткової.

**Кінцевий результат:** Кінцевим результатом даної роботи є математичне та програмне забезпечення, архітектура моделі, вагові коефіцієнти натренованої моделі та код програмного забезпечення, в якому реалізовано дану роботу.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

- \* Модель – нейронна мережа
- \* Система (система для маркування зображень) – композиція моделей
- \* Задача класифікації – це задача, яка вирішує проблему приналежності чогось виключно до одного класу із довільного набору класів.
- \* Задача маркування – це задача, яка вирішує проблему приналежності чогось до декількох класів із довільного набору класів.
- \* DNN – Глибинна нейронна мережа (Deep Neural Network)
- \* CNN – Згорткова нейронна мережа (Convolutional Neural Network)
- \* RNN – Рекурсивна нейронна мережа (Recursive Neural Network)
- \* Тег (Tag) – шумна інформація надана користувачем у формі тексту (наприклад для зображення кота "Cat, Canada, Cola")
- \* Лейбл (Label) – синонім слова ground truth для класифікації

## ЗМІСТ

Перелік умовних позначень, скорочень і термінів .....	4
1 Вступ .....	6
2 Огляд існуючих рішень .....	8
3 Моделювання .....	11
3.1 VCNN .....	11
3.2 MLP .....	12
3.3 LP .....	12
3.4 LQP .....	13
3.5 Процес тренування .....	13
3.6 Процес тестування .....	16
4 Експерименти .....	17
4.1 Датасет .....	17
4.2 Метрики .....	19
4.3 Тренування .....	21
5 Аналіз результатів .....	23
5.1 Аналіз компонентів системи .....	23
5.2 Порівняння з існуючими рішеннями .....	24
5.3 Демонстративні приклади .....	26
Висновки .....	28
Перелік посилань .....	29
Додаток А Код лістинг .....	31
Додаток Б Додаткові приклади .....	32

## 1 ВСТУП

Задача класифікації – це одна із основних задач в аналізі зображень, вона полягає у присвоєнні кожному зображенню один із класів. Таким чином дане формулювання накладає обмеження – зображення містить тільки один об'єкт. Поява DNN [3] та її подальший розвитком у CNN [14, 13] разом із створенням великих датасетів як-от ImageNet [4] дало змогу вирішувати задачу класифікації зображень значно швидше і якісніше ніж люди.

Зрозуміло, що зображення – це той тип даних, який у абсолютній більшості випадків містить більше одного об'єкта, і відповідно більше одного класу для класифікації. Для поглиблення опису існує задача маркування зображень (image labeling). На відміну від класифікації, вона полягає у маркуванні зображення більше ніж одним класом. Таким чином повнота опису зображення кратно зростає у порівнянні із звичайною класифікацією, однак привносить декілька складних завдань.

По-перше, наявність декількох класів у одного зображення створює можливість описувати значно ширший спектр візуальної інформації: різні об'єкти, стилі, дії, і тд. Це створює потребу у розгляді додаткових джерел інформації, так як одного лиш зображення вже недостатньо. Поява великих хостингів зображень таких як Imgur, Flickr, та ін., де користувачі можуть як завантажувати різноманітні зображення, так і додавати до них описову інформацію у вигляді тегів / анотацій, дала змогу створити досить різноманітні датасети: ImageNet [4], MS-COCO [9], NUS-WIDE [2], та ін. Також існують і інші види датасетів, наприклад: рентгенівські знімки та додаткова інформація (інші аналізи пацієнта, історія хвороб ...), супутникові знімки та додаткова інформація у вигляді метаданих, геолокацій тощо. Таким чином задача якісного маркування зображення вже охоплює значно більший спектр даних ніж просто зображення.

По-друге, маркування зображень передбачає динамічну к-сть промаркованих класів, так для опису зображенням із широким спектром

понять необхідно 5-6 класів, для зображення із простим вмістом - 2-3 класи.

По-третє, маркування зображень потребує оцінки якості проведеного маркування. Оскільки будь який датасет буде містити в собі дизбаланс класів в тій чи іншій мірі, важливо оцінювати маркування із урахуванням цього.

По-четверте, маркування зображень значно складніша задача ніж класифікація і відповідно зростає складність моделей. З однієї сторони складніша модель потенційно здатна покращити якість маркування, з іншої - може сильно збільшити як час на виконання маркування, так і час затрачений на тренування системи. До складних систем належать ті, які використовують трансформери та/або мають велику к-сть параметрів. Отже, важливо обрати певний баланс відносно складності моделі.

Все це робить задачу маркування зображення досить складною.

## 2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Розглянемо ключові аспекти задачі маркування зображення.

### **Базове рішення**

Базовим рішенням в задачі маркування зображення є аналіз основної модальності даних – зображення. У абсолютній більшості існуючих робіт використовується CNN (Convolutional Neural Network). Застосовуються різні архітектури даної моделі ResNet [6], AlexNet [1], GoogleNet [16], ResNext [19]. Спільним між ними є те, що вони вже натреновані на великому датасеті, здебільшого ImageNet [4]. Для адаптації моделі до обраного контексту така модель дотреновується (fine tune), замінюючи існуючий класифікатор на класифікатор із адаптованою к-стю вихідних класів [5]. Це працює завдяки тому, що всі архітектури сучасних CNN моделей є багат шаровими, і перші шари розпізнають базові особливості (features) зображення, а останні шари вивчають більш глибокі особливості зображення, таким чином дозволяючи адаптовувати модель під різні задачі після проведення підгонки (finetune).

### **Додаткова модальність даних**

Більш нові роботи також розглядають додаткові джерела інформації для підвищення якості та повноти маркування зображень. Існує два основних підходи:

а) *Аналіз додаткової інформації.* Даний підхід аналізує додаткову до зображення інформацію. Це може бути як текстова інформація (теги / анотації) [7, 10], так і метадані зображення [8, 17]. Очевидним недоліком даного методу є потреба у цій додатковій інформації, яку можуть мати не всі зображення, а відсутність даної інформації знижує точність результуючого маркування.

б) *Аналіз цільових класів.* На відміну від загальної інтерпретації



класів для задачі маркування (коли кожен клас – це незалежна сутність), даний підхід аналізує зв'язки між цільовими класами, створюючи нову модальність на основі набору цільових класів [22]. Більш новим та узагальненим підходом є технологія word2vec (або аналогічне рішення), яка дозволяє, проаналізувавши набір слів, впорядкувати його у векторному просторі за семантичним значенням і побудувати граф, в якому вказуються класи (поняття) які часто знаходяться на одному зображенні (co-occurrence; наприклад: риба, вода) та класи які рідко знаходяться на одному зображенні (dis-occurrence; наприклад: риба, пустеля) [11, 20, 21]. Перевагою даного підходу є те, що йому не потрібні ніякі нові дані крім зображень та відповідних їм цільових класів, а нова модальність, яка репрезентує зв'язок між класами створюється під час тренування системи. Основним недоліком таких систем полягає у високій складності, і як наслідок – довше по часу тренування / розпізнавання.

### **К-сть лейблів**

Результатом роботи класифікаційної моделі є вектор ймовірностей, який репрезентує приналежність до класів. Для задачі класифікації вибір результату на основі цього вектора очевидний – клас із найбільшою ймовірністю, однак для задачі маркування все складніше. Більшість наведених вище робіт розглядають задачу вибору  $k$ -сті лейблів як найкращі  $k$  (top  $k$ ) маркувань (Розділ 4.2), де  $k$  – наперед задана константа. Очевидно, що такий вибір  $k$ -сті класів не є оптимальним, так як більш змістовні зображення будуть містити менше описової інформації і навпаки – менш змістовні будуть містити лишню інформацію, яка до того ж може не мати нічого спільного із цим зображенням (Рис. 2.1)

Один із підходів як-от CNN-RNN [18], використовує RNN для аналізу візуальних даних (visual features) та автоматично виконує як задачу маркування, так і задачу динамічного вибору кількості лейблів, однак в силу особливості RNN є певні обмеження накладенні на порядок кодування класів.





Image				
Truth	clouds grass house sky	leaf plants sky	animal grass	person
Top 5 pred	clouds grass house road sky	clouds grass leaf plants sky	animal grass horses plants sky	animal clouds person road sky
Model pred	clouds grass sky	plants sky	animal grass horses	person

Рис. 2.1 – Приклад адаптивної к-сті лейблів. 'Truth' – правдиве маркування, 'Top 5 pred' – ілюстрація вибору top  $k$ , при  $k = 5$ , 'Model pred' – приклад роботи моделі

Найновіші моделі [11, 20, 21], які розглядають зв'язок між цільовими класами, обирають к-сть цільових класів на основі порогового значення (Розділ 4.2). Варто зазначити, що даний підхід є ефективним тільки для цього типу моделей, так як вектор ймовірностей, який отримується на виході даної моделі досить сильно дискретизований, тобто для позитивного лейблу, який маркується 1 ймовірність буде  $\approx 0.7 - 0.9$ , а для негативного, тобто 0 ймовірність  $\approx 0.1 - 0.3$ .

### 3 МОДЕЛЮВАННЯ

На основі проведеного аналізу альтернатив, дана робота пропонує розглянути модель, яка розглядає дві модальності даних: зображення та текстові теги. Архітектура даного рішення складається із чотирьох компонентів (Рис. 3.1).

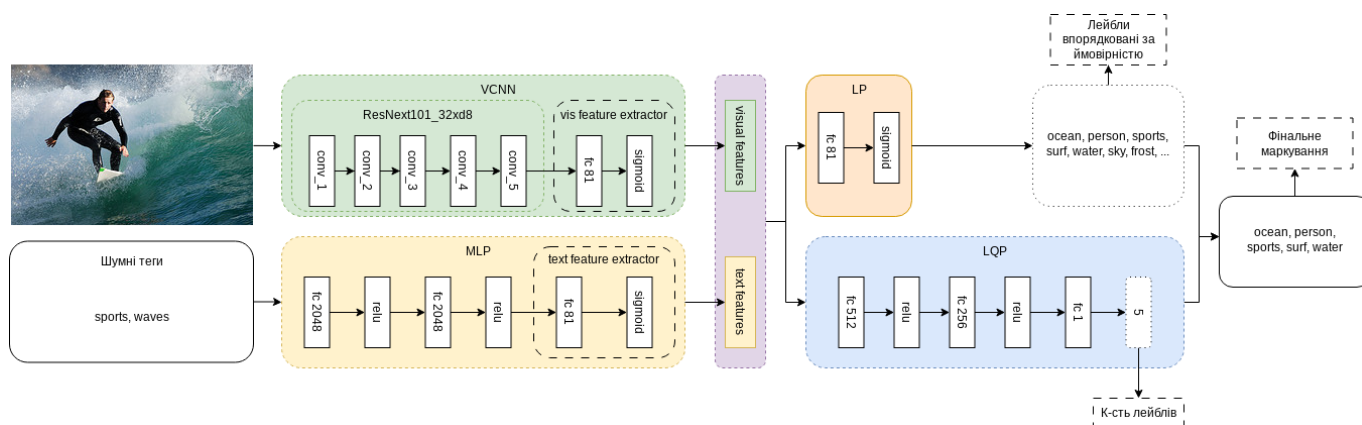


Рис. 3.1 – Архітектура композитної системи

#### 3.1 VCNN

Модель VCNN (Рис. 3.1) призначена для вивчення особливостей (features) із зображення. Отримує на вхід пісели зображення  $I$ , у формі матриці розмірності  $(B, C, W, H)$ , де  $B$  – к-сть зображень у групі для тегування,  $C$  – к-сть каналів у зображеннях зазвичай 1 або 3, Grey або RGB відповідно,  $W, H$  – розмірність зображень.

За базове рішення використовується ResNext101\_32x8d [19] (сучасна версія resnet), із адаптованим класифікатором (vis feature extractor), натреновану на датасеті ImageNet [4].

На виході даної моделі ми отримуємо вектор вірогідностей  $vf$  (visual feature vector), який вказує вірогідність маркування зображення класом  $j$  на основі візуальної інформації.

### 3.2 MLP

MLP (Рис. 3.1) – аналізує текстові особливості (text features) тегів до зображення. Теги до зображення  $i$  репрезентуються як бінарний вектор  $I = [1, 0, 1, 0, \dots, N]$ , де 1 – це наявність тегу, а  $N$  – к-сть тегів.

Головна причина вибору звичайної MLP моделі для аналізу текстової інформації – це те, що вхідна інформація – це шумні теги (наприклад: для фото кота – теги "Канада", "Kit").

На виході даної моделі ми отримуємо вектор  $tf$  вірогідностей (text feature vector), який вказує вірогідність маркування зображення класом  $j$  на основі текстової інформації.

### 3.3 LP

LP (Рис. 3.1) – аналізує вектор вірогідності  $v$ , який є композицією векторів  $vf$  та  $tf$ :  $v = [vf, tf]$ .

На виході даної моделі ми отримуємо вектор вірогідностей, який комбінує інформацію отриману як із візуальної так і з текстової інформації.

### 3.4 LQP

Модель LQP (Рис. 3.1) аналізує кількість лейблів на основі вектору вірогідностей  $v$ , який є композицією векторів  $vf$  та  $tf$ :  $v = [vf, tf]$ .

Існує два підходи до визначення к-сті за допомогою нейронних мереж: класифікація та регресія. LQP – регресійна модель.

Оскільки регресійні моделі досить швидко перенавчаються (overfitting), то необхідно задіяти регуляризацію. В даній роботі, в якості регуляризатора задіяні Dropout шари [15], із вірогідністю відкидання (dropout rate) 0.5.

На виході даної моделі є число, яке вказує на кількість лейблів у зображенні.

### 3.5 Процес тренування

Модель складається із декількох компонентів, що створює декілька проблемних місць під час тренування: досить багато параметрів, дві різні цільові функції, проблема затухаючого градієнта, – тому тренування відбувається у декілька стадій, у якому кожна із моделей тренується окремо (деякі з них можна тренувати синхронно).

#### Цільові функції

Для початку варто розглянути цільові функції (функція втрат, loss function). Дані функції є базовим компонентом глибокого навчання.

В даній роботі використовуються дві функції: BCEWithLogitsLoss та MSELoss.

### *BCEWithLogitsLoss*

Для тренування класифікаційних моделей (VCNN, MLP, LP) вихідні логіти  $z_{ij}$  для групи (batch) зображень  $I_N$  при  $i = 1...N$ ,  $j = 1...C$ , де  $N$  - кількість зображень в групі,  $C$  - кількість цільових класів, цільова функція має вигляд:

$$\mathcal{L}_{cls} = \frac{1}{NC} \sum_i^N \sum_j^C y_{ij} \cdot \ln(\sigma(z_{ij})) + (1 - y_{ij}) \cdot \ln(1 - \sigma(z_{ij})) \quad (3.1)$$

, де  $y_{ij} = 1$  якщо зображення  $i$  анотоване класом  $j$ , інакше -  $y_{ij} = 0$ , а  $\sigma(\cdot)$  - це сигмоїдальна активаційна функція

### *MSELoss*

Для тренування регресійної моделі LQP вихідні логіти  $z_i$  для групи (batch) зображень  $I_N$  при  $i = 1...N$ , де  $N$  - кількість зображень в групі, цільова функція має вигляд:

$$\mathcal{L}_{reg} = \frac{1}{N} \sum_i^N (y_i - z_i)^2 \quad (3.2)$$

, де  $y_i$  - це кількість лейблів для зображення  $I_i$ .

### **Тренування VCNN**

Тренування моделі ResNext [19] з нуля є досить складною задачею (дана модель має  $\approx 80M$  параметрів), адже для цього потрібні значні обчислювальні потужності.

Саме тому виристовується натренована модель із адаптованим класифікатором (visual feature extractor) (Рис. 3.1), яка підганяється

(finetuned) на обраному датасеті.

Існує два підходи для підгонки:

1) Підгонка всієї моделі: всі шари моделі підганяються (дотреновуються) з низькою швидкістю навчання (learning rate). Даний підхід вимагає великої обчислювальної потужності, однак надає високу точність та досить таки швидко тренується (в порівнянні із тренуванням з нуля).

2) Підгонка класифікатора: відбувається тренування тільки класифікатора, фіксуючи всі інші параметри моделі. Даний підхід значно пришвидшує тренування в обмін на певну деградацію точності в порівнянні із 1-им варіантом.

В даній роботі використовується 1 варіант підгонки, так як він надає вищу точність.

### **Тренування MLP**

Дана модель є звичайним багат шаровим персептроном, тому її тренування досить таки прямолінійне.

### **Тренування LP**

Дана модель призначена для обрахування вірогідностей на основі вектору  $f = [vf, tf]$ , оскільки вона складається із одного шару то її тренування також очевидне.

### **Тренування LQP**

Дана модель є регресійною, її тренування також є очевидним, однак варто нормалізувати вхідну к-сть лейблів, так як це пришвидшить та/або покращить збіжність моделі.

### 3.6 Процес тестування

#### Тестування класифікаційних моделей (VCNN, MLP, LP)

Кожна із даних моделей обраховує вектор ймовірностей  $P$ , для тестування необхідно перевести вектор ймовірностей (наприклад:  $P = [0.9, 0.6, 0.1, 0.4, 0.6]$ ) у вигляд маркування (наприклад:  $M = [1, 1, 0, 0, 1]$ ). Дане перетворення називається індикаторною функцією.

Розглянемо два основних види індикаторної функції:

1) Порогове значення (threshold): для вектору ймовірностей  $P$  та порогу  $\alpha$  – вектор маркувань обраховується наступним чином: якщо  $y_i > \alpha$ , то маркуємо 1, інакше 0. Наприклад при  $\alpha = 0.5$ :  $[0.9, 0.6, 0.1, 0.4, 0.6] \rightarrow [1, 1, 0, 0, 1]$ .

2) Найкращі  $k$  (top  $k$ ): для вектору ймовірностей  $P$  та числа  $k$  – вектор маркувань обраховується наступним чином: маркуємо 1 найкращі  $k$  ймовірностей, інакше 0. Наприклад при  $k = 4$ :  $[0.9, 0.6, 0.1, 0.4, 0.6] \rightarrow [1, 1, 0, 1, 1]$ .

Оскільки дані моделі **не передбачають** передбачення кількості лейблів, то для тренування даних моделей використовується метод top  $k$ , при чому  $k = 3$ .

Тобто для зображення  $I$ , із маркуванням  $Y = [1, 0, 0, 0, 1]$ , і вектором ймовірностей  $P = [0.8, 0.9, 0.1, 0.5, 0.2]$  та результуючим вектором маркувань  $M$ :  $k = 3$ , перетворення  $P \rightarrow M \equiv [0.8, 0.9, 0.1, 0.3, 0.2] \rightarrow [1, 1, 0, 1, 0]$

#### Тестування композитної моделі

Для тестування композитної моделі (Рис. 3.1) необхідно обрахувати результуючі значення для моделей LP та LQP, і обрати top  $k$  лейблів LP, де  $k$  – це передбачення LQP.



## 4 ЕКСПЕРИМЕНТИ

## 4.1 Датасет

Один із найбільш часто використовуваних датасетів для тестування моделей маркування зображень – NUS-WIDE [2], він складається із 269,655 зображень, 81 лейблу, та  $\approx 5000$  тегів в якості сторонньої текстової інформації. Для проведення тренування/тестування використовується розподіл, надведений разом із датасетом, так як він є збалансований настільки, наскільки це можливо (Рис. 4.1).

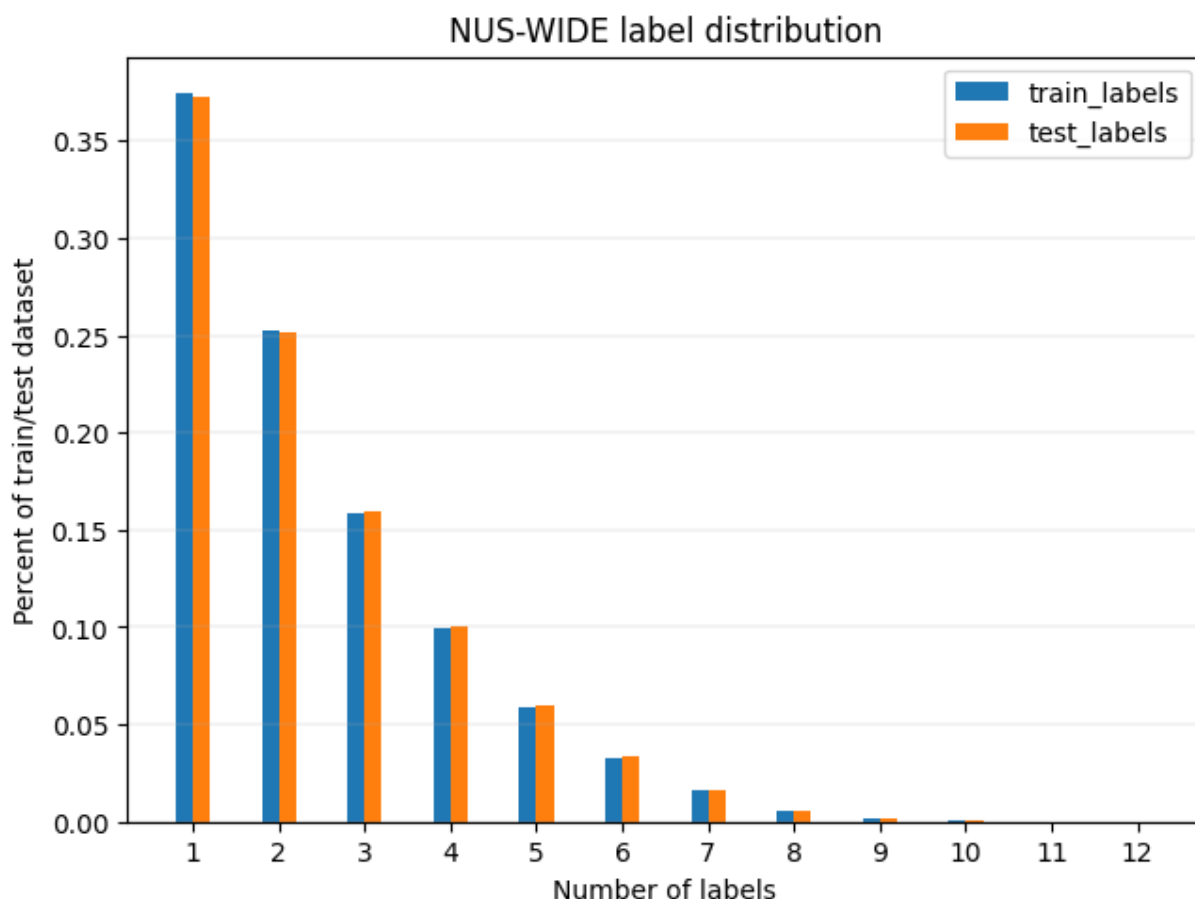


Рис. 4.1 – Розподіл лейблів у тренувальному/тестовому датасеті

Важливо відмітити, що даний датасет містить посилання на зображення на ресурсі Flickr, і деякої частина цих зображень вже там немає. Також

буде використано тільки 1000 найбільш частих тегів з  $\approx 5000$ , при чому зображення, які не містять жодного тега відфільтровано.

Далі наведено деякі числові характеристики тренувальної/тестової вибірок:

	Тренування	Тестування
Кількість зображень	121962	81636
Середня к-сть лейблів	2.42	2.43
Медіана к-сть лейблів	2	2
Мінімальна к-сть лейблів	1	1
Максимальна к-сть лейблів	12	13

Табл. 4.1 – Характеристики тренувальної/тестової вибірок

## 4.2 Метрики

Для оцінки точності будуть використовуватись метрики, які є загально вживаними для оцінки задачі маркування зображень (multi-label image annotation).

$$\begin{aligned}
 C-P &= \frac{1}{C} \sum_{j=1}^C \frac{NI_j^c}{NI_j^p} & O-P &= \frac{\sum_{i=1}^N NL_i^c}{\sum_{i=1}^N NL_i^p} \\
 C-R &= \frac{1}{C} \sum_{j=1}^C \frac{NI_j^c}{NI_j^g} & O-R &= \frac{\sum_{i=1}^N NL_i^c}{\sum_{i=1}^N NL_i^g} \\
 C-F1 &= \frac{2 \cdot C-P \cdot C-R}{C-P + C-R} & O-F1 &= \frac{2 \cdot O-P \cdot O-R}{O-P + O-R}
 \end{aligned} \tag{4.1}$$

,де

- \*  $C$  - к-сть класів
- \*  $N$  - к-сть тестових зображень
- \*  $NI_j^c$  - к-сть зображень які **коректно** промарковано як клас  $j$
- \*  $NI_j^g$  - к-сть зображень які мають клас  $j$
- \*  $NI_j^p$  - к-сть зображень які промарковано як клас  $j$
- \*  $NL_i^c$  - к-сть **коректно** промаркованих лейблів для зображення  $i$
- \*  $NL_i^g$  - к-сть лейблів які має зображення  $i$
- \*  $NL_i^p$  - к-сть промаркованих лейблів для зображення  $i$

Варто відзначити, що дані метрики є зміщеними (biased), при чому по-класові метрики (C) зміщені в сторону рідкісних класів, а загальні метрики (O) - в сторону частих класів [5].

Для того щоб отримати уніфіроване представлення про ефективність моделі буде використовуватись наступна метрика, яка бере до уваги як C-F1 так і O-F1, що полегшує інтерпретацію результатів:

$$H-F1 = \frac{2 \cdot C-F1 \cdot O-F1}{C-F1 + O-F1} \quad (4.2)$$

### 4.3 Тренування

Для програмної реалізації запропонованої моделі було використано PyTorch.

Оскільки в даній роботі, використовується модель ResNext [19] натренована на датасеті ImageNet [4], то для вхідних зображень потрібно застосувати певне перетворення:

1) Зміна розміру (Resize)  $232 \times 232$ , використовуючи білінійну інтерполяцію

2) Центральний кроп (Central crop)  $224 \times 224$

3) Зміна масштабу (Rescale)  $[0,1]$

4) Нормалізація на основі статистичних величин ImageNet [4]. А саме:  $\text{mean} = [0.485, 0.456, 0.406]$  та  $\text{std} = [0.229, 0.224, 0.225]$

Дане перетворення доступно у бібліотеці PyTorch.

#### Параметри навчання

Класифікаційні моделі VCNN та MLP навчалися із швидкістю навчання (learning rate) 0.001, а LP – зі швидкістю 0.01. Також для навчання цих трьох моделей використовувався контролер швидкості навчання (learning rate scheduler), який множив швидкість навчання на 0.5, досягаючи 5-ої та 10-ої епохи.

Регресійна модель LQP навчалась із сталою швидкістю навчання (learning rate) 0.0005.

Для всіх навчання всіх вище згаданих моделей використовувався оптимізатор AdamW, із параметром l1 регуляризації (weight decay) 0.0003.

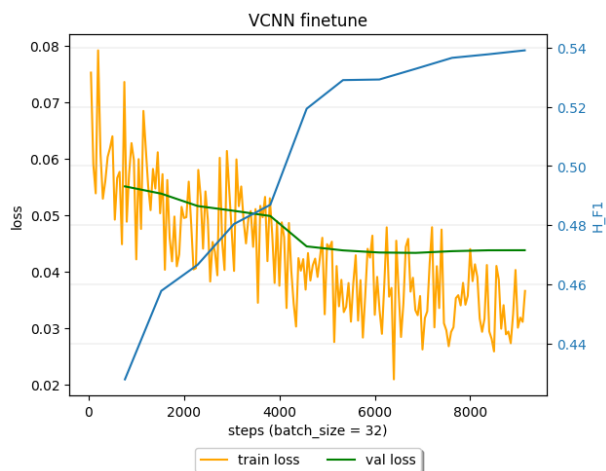
Розмір групи (batch size) 32.

Також варто відзначити, що в даній роботі епоха – це 20% від усіх даних, при чому після кожної епохи дані перемішуються (shuffle), отримавши нові 20% даних.

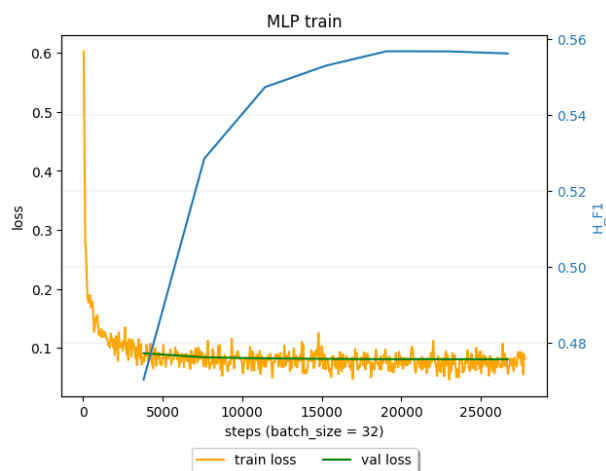
## Процес навчання

Для тренування було використано графічний процесор 'Nvidia L4'. Для тренування всіх елементів моделі знадобилось  $\approx 3$  години.

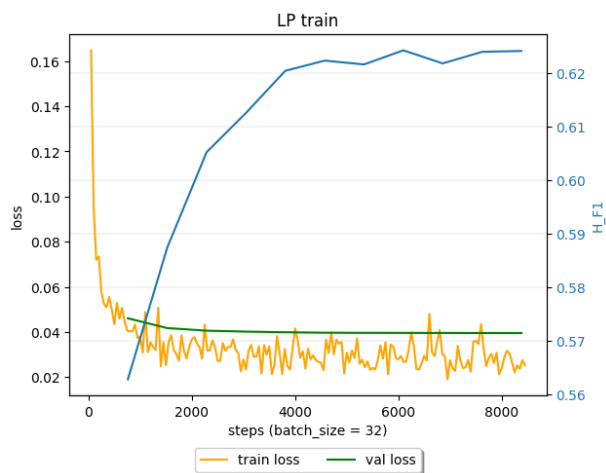
Для оптимізації процесу тренування було застосовано техніку mixed precision, яка використовує f16 замість f32, під час певних етапів тренування [12].



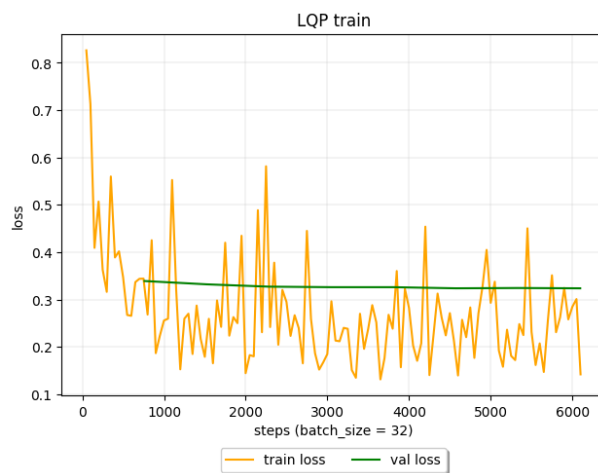
(4.2a) Навчання VCNN



(4.2б) Навчання MLP



(4.2в) Навчання LP



(4.2г) Навчання LQP

Рис. 4.2 – Процес навчання моделей

## 5 АНАЛІЗ РЕЗУЛЬТАТІВ

## 5.1 Аналіз компонентів системи

Модель	Індикаторна ф-ція (3.6)	Модальність	C-P	C-R	C-F1	O-P	O-R	O-F1	H-F1
Композитна	top k	Зображення+теги	72.49	59.51	65.36	76.53	74.41	75.46	70.04
VCNN+MLP+LP	top 3	Зображення+теги	60.51	61.28	60.89	67.87	71.52	63.98	62.40
VCNN+LQP	top k	Зображення	64.62	37.61	47.54	77.07	59.01	66.84	55.56
VCNN	top 3	Зображення	44.48	53.32	48.50	55.44	68.52	61.29	54.15

Табл. 5.1 – Порівняння компонентів моделі

Результуюча композитна модель показала значно кращий результат ніж базове рішення (VCNN).

**Додаткова модальність (MLP+LP)**

Додавання додаткової модальності у вигляді тегів внесло значний вклад у підвищення якості маркування. Порівнюючи відповідні метрики H-F1 для моделей VCNN та VCNN+MLP+LP, можна побачити приріст на 8.25%. При чому варто відзначити, що цей ріст в основному забезпечений приростом метрики C-F1, яка є зміщеною в сторону більш рідкісних класів. Варто відзначити, що це працює завдяки тому що зазвичай теги надані користувачами відмічають досить рідкісні поняття, які на відміну від частих тегів (небо, сонце, людина, вода і тд.) складно розпізнати маючи одне лиш зображення.

**Передбачення кількості лейблів (LQP)**

При використанні компоненту LQP кількість лейблів обирається за принципом 'top k', а не 'top 3' (3.6). Це очевидним чином підвищує точність фінального маркування, адже деякі зображення можуть мати більше трьох лейблів, інші – менше трьох. Порівнюючи вплив компоненти LQP для базового рішення (VCNN)

та композитної моделі (VCNN+MLP+LP+LQP) можна зробити припущення, що VCNN аналізує загальні поняття на зображенні, а враховуючи дисбаланс класів у датасеті, використання принципу 'top k' збільшує точність (precision), сильно жертвуючи по-класовим охопленням (C-R), і, як наслідок, не сильно збільшуючи величину головної метрики H-F1. На практиці це виливалось у те, абсолютна більшість зображень маркувалась частими лейблами (людина, вода, небо і тд.), а рідкісні теги - ігнорувались. Натомість у композитній моделі вищезгаданий принцип чудово проявив себе. Згідно із тестовими метриками покращення становить 7.64%.

## 5.2 Порівняння з існуючими рішеннями

Модель	Індикаторна ф-ція (3.6)	Модальність	C-P	C-R	C-F1	O-P	O-R	O-F1	H-F1
Композитна	top k	Зображення+теги	72.49	59.51	65.36	76.53	74.41	75.56	70.04
Query2Label [11]	threshold $\alpha$	Зображення (+аналіз класів)	-	-	67.60	-	-	76.3	71.69
SR-CNN-RNN [10]	top 3	Зображення+теги (+аналіз класів)	71.73	61.73	66.36	77.41	76.88	77.15	71.35
Resnet-CPSD [20]	threshold $\alpha$	Зображення (+аналіз класів)	-	-	64.00	-	-	75.30	69.19
MS-CMA [21]	threshold $\alpha$	Зображення (+аналіз класів)	-	-	60.50	-	-	73.80	66.49
Resnet-SRN [22]	threshold 0.5	Зображення (+аналіз класів)	65.20	55.80	58.50	75.50	71.50	73.40	65.10
SINN [7]	top 3	Зображення+теги	58.30	60.63	59.44	57.05	79.12	66.29	62.68
TagNeighbour [8]	top 3	Зображення+метадані	54.74	57.30	55.99	53.46	75.10	62.46	59.05
CNN+Logistic [7]	top 3	Зображення	45.60	45.03	45.31	51.32	70.77	59.50	51.44
CNN-RNN [18]	top 3	Зображення	40.50	30.40	34.70	49.9	61.70	55.20	42.61
CNN+WARP [5]	top 3	Зображення	31.65	35.60	33.51	48.59	60.49	53.89	41.32
CNN+Softmax [5]	top 3	Зображення	31.68	31.22	31.45	47.82	59.52	53.03	39.48

Табл. 5.2 – Порівняння результуючих метрик для різних моделей на датасеті NUS-WIDE



**Точність запропонованого рішення** Композитна модель (VCNN+MLP+LP+LQP) продемонструвала високу якість маркування на тестових метриках у порівнянні із розглянутими альтернативними рішеннями. Згідно із метрикою H-F1 запропоноване рішення є третім.

**Модальність даних** Задача маркування зображень розглядає зображення як основну модальність, однак додавання модальності, очікувано, покращує результати маркування. Це підтверджують метрики наведені в Табл. 5.2. Серед розглянутих рішень є 3 варіанти модальності даних з якимим працюють нейронні мережі. Найменш ефективним, як і очікувалось, виявились моделі які аналізують виключно зображення. Введення інших двох видів додаткових модальностей: теги, аналіз класів, - надають значно кращі результати. Варто відзначити, - аналіз класів (найкраще імплементовано в: Query2Label [11], Resnet-CPSD [20] та MS-CMA [21]) не потребує ніяких додакових даних окрім зображення, що є вагомою конкурентною перевагою, враховуючи незначну відмінність в точності моделей.

**Індикаторна функція** Для оцінки ефективності запропонованої індикаторної функції 'top k', варто ізолювати вплив саме цієї фнкції. Для цього розглянемо існуючі моделі, які працюють із тією ж модальністю даних. Найкращою із таких моделей є SINN [7]. Використання під-системи LQP, яка передбачає роботу із динамічною кількістю лейблів при маркуванні (top k) значено підвищує якість. Згідно із наведеними метриками покращення складає 7.36% (Табл. 5.2), що є вагомим приростом.

### 5.3 Демонстративні приклади

З тестового датасету випадковим чином обрано декілька зображень, для демонстрації роботи моделі:















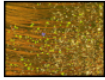

Image	Truth	Model pred	Image	Truth	Model pred
	clouds sky	clouds sky		snow water	glacier lake water
	flowers plants	flowers plants		clouds mountain rocks sky	mountain rocks sky
	cityscape clouds sky	buildings clouds nighttime sky		buildings clouds	flowers
	clouds	clouds sky		clouds person sky	buildings person sky
(5.1a)			(5.1б)		
Image	Truth	Model pred	Image	Truth	Model pred
	clouds ocean water	clouds military sky		person wedding	person wedding
	person	person sky		bear	animal
	clouds moon ocean sky water	lake moon ocean sky water		clouds nighttime sky window	buildings clouds nighttime sky
	flowers grass water	grass plants		flowers garden grass plants water	flowers garden grass plants
(5.1в)			(5.1г)		

Рис. 5.1 – Демонстративні приклади

Задамо умовне позначення "а::1", що означає демонстративний приклад "а", перше зображення зверху, "в::2" – приклад "в", друге зображення зверху і тд.

Серед наведених прикладів можна розглянути кілька цікавих моментів, які не відображають тестові метрики:

\* Іноді модель передбачає маркування, якого немає у датасеті, однак присутнє на зображені. Наприклад: [а::3,а::4,б::1,б::3,б::4,в::1,в::2,г::2,г::3].

\* Існують випадки коли модель відмічає поняття, які не можуть бути присутніми на одному зображенні. Це є прямим наслідком того, що наша модель розглядає цільові класи як незалежі сутності. Наприклад: lake та ocean як-от в прикладі 'в::3'.

\* Іноді передбачення моделі відсікають неіснуючі поняття та маркують зображення краще ніж це було зроблено в датасеті. Так для зображення 'б::3' на якому зображено якусь рослину датасет вказує що це: 'buildings, clouds'; а модель - 'flowers'.

Окрім наведених вище особливих випадків, іноді модель, звичайно, помиляється. Однак у наведених прикладах немає значних помилок у маркуванні.

## ВИСНОВКИ

В даній роботі було розглянуто композитну модель, для маркування зображень (шпалерів робочого столу) проведено оцінювання її точності на тестових метриках.

Розглянута модель показала хороший результат у порівнянні із існуючими альтернативними рішеннями.

*До переваг розглянутого рішення належать:*

- + Висока точність
- + Невелика кількість параметрів ( $\approx 95\%$  параметрів має модель для аналізу зображень)
- + Висока швидкість тренування

*Недоліками є:*

- Неможливість тренування моделі в один етап (end-to-end)
- Необхідність використання додаткових даних (тегів) для отримання високої якості опису зображення

В подальшому варто розглянути ефективність даної моделі на інших датасетах (наприклад MSCOCO [9]). також варто розглянути і методи для аналізу цільових класів, так як незважаючи на значне ускладнення фінальної моделі це надає досить високий приріст до точності маркування.

Результуюча композитна модель збережена у форматі safetensors.

Після проведеного дослідження було висунуто гіпотезу, що дане рішення можна застосувати і у інших схожих предметних областях. Так для аналізу рентгенівських знімків - це може бути історія хвороб пацієнта, для супутникових знімків - різні метадані, геолокація тощо, а для аналізу звичайних фотографій - теги, анотації, метадані, тощо.

## ПЕРЕЛІК ПОСИЛАНЬ

- [1] Krizhevsky Alex, Sutskever Ilya та Hinton Geoffrey. „ImageNet Classification with Deep Convolutional Neural Networks“. В: *Advances in Neural Information Processing Systems*. За ред. F. Pereira та ін. Т. 25. Curran Associates, Inc., 2012. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf).
- [2] Tat-Seng Chua та ін. „NUS-WIDE: A Real-World Web Image Database from National University of Singapore“. В: *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*. Santorini, Greece., July 8-10, 2009.
- [3] Dan Cireşan, Ueli Meier та Juergen Schmidhuber. „Multi-column deep neural networks for image classification“. В: (лют. 2012). arXiv: 1202.2745 [cs.CV].
- [4] Li Deng та ін. „Imagenet: A large-scale hierarchical image database“. В: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, С. 248—255.
- [5] Yunchao Gong та ін. *Deep Convolutional Ranking for Multilabel Image Annotation*. 2013. eprint: arXiv:1312.4894.
- [6] Kaiming He та ін. „Deep residual learning for image recognition“. В: (груд. 2015). arXiv: 1512.03385 [cs.CV].
- [7] Hexiang Hu та ін. „Learning structured inference neural networks with label relations“. В: (листоп. 2015). arXiv: 1511.05616 [cs.CV].

- [8] Justin Johnson, Lamberto Ballan та Fei-Fei Li. „Love thy neighbors: Image annotation by exploiting image metadata“. В: (серп. 2015). arXiv: 1508.07647 [cs.CV].
- [9] Tsung-Yi Lin та ін. „Microsoft COCO: Common Objects in Context“. В: *CoRR* abs/1405.0312 (2014). arXiv: 1405.0312. URL: <http://arxiv.org/abs/1405.0312>.
- [10] Feng Liu та ін. „Semantic Regularisation for Recurrent Image Annotation“. В: (листоп. 2016). arXiv: 1611.05490 [cs.CV].
- [11] Shilong Liu та ін. „Query2Label: A Simple Transformer Way to Multi-Label Classification“. В: (лип. 2021). arXiv: 2107.10834 [cs.CV].
- [12] Paulius Micikevicius та ін. „Mixed Precision Training“. В: (жовт. 2017). arXiv: 1710.03740 [cs.AI].
- [13] Keiron O'Shea та Ryan Nash. „An Introduction to Convolutional Neural Networks“. В: (листоп. 2015). arXiv: 1511.08458 [cs.NE].
- [14] Karen Simonyan та Andrew Zisserman. „Very deep convolutional networks for large-scale image recognition“. В: (вер. 2014). arXiv: 1409.1556 [cs.CV].
- [15] Nitish Srivastava та ін. „Dropout: A Simple Way to Prevent Neural Networks from Overfitting“. В: *Journal of Machine Learning Research* 15.56 (2014), С. 1929—1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html>.
- [16] Christian Szegedy та ін. *Going Deeper with Convolutions*. 2014. eprint: arXiv:1409.4842.

- [17] Kevin Tang та ін. „Improving image classification with location context“. В: (трав. 2015). arXiv: 1505.03873 [cs.CV].
- [18] Jiang Wang та ін. „CNN-RNN: A unified framework for multi-label image classification“. В: (квіт. 2016). arXiv: 1604.04573 [cs.CV].
- [19] Saining Xie та ін. „Aggregated residual transformations for deep neural networks“. В: (листоп. 2016). arXiv: 1611.05431 [cs.CV].
- [20] Jiazhi Xu та ін. „Boosting multi-Label Image Classification with complementary Parallel Self-distillation“. В: (трав. 2022). arXiv: 2205.10986 [cs.CV].
- [21] Renchun You та ін. „Cross-modality attention with semantic graph embedding for multi-label classification“. В: (груд. 2019). arXiv: 1912.07872 [cs.CV].
- [22] Feng Zhu та ін. „Learning spatial regularization with image-level supervisions for multi-label image classification“. В: (лют. 2017). arXiv: 1702.05891 [cs.CV].

## Додаток А

### Код лістинг

Код міститься у github репозиторії [Посилання](#)

Далі наведено загальний опис елементів проекту:

#### **Дані:**

Файли в директорії `scripts` `'nuswide2ndjson.py'` та `'1ktags.py'` призначенні для обробки сирих даних із датасету в формат `ndjson` для подальшого тренування.

Файл `data.py` містить адаптери та визначення датасету для тренування.

#### **Тренування:**

Скрипти для тренування для всіх моделей (VCNN, MLP, LP, LQP) знаходяться в директорії `'scripts/train'`.

#### **Тестування:**

Скрипт для тестування моделей: `'scripts/test.py'`. Даний скрипт передбачає тестування як фінальної моделі, так і деяких конфігурацій її компонентів.

#### **Інше:**

Скрипт `'scripts/compose2safe.py'` призначений для конвертації вагів моделей (VCNN, MLP, LP, LQP) формату `.skpt` у композитну модель формату `.safetensors`.

Ноутбук `'testing.ipynb'` призначений для наглядного тестування моделі.



Додаток В

Додаткові приклади





Image	Truth	Model pred
	animal birds sky	animal birds sky
	person wedding	person wedding
	buildings	sky
	animal	animal birds

Рис. В.1





Image	Truth	Model pred
	animal clouds	animal
	animal	animal
	flowers	flowers
	animal elk snow	animal snow

Рис. В.2





Image	Truth	Model pred
	sky	animal clouds horses sky
	train	railroad train
	buildings clouds sky window	buildings clouds sky town window
	clouds grass sky vehicle	clouds sky vehicle window

Рис. В.3

Image	Truth	Model pred
	clouds sky	buildings clouds grass sky
	buildings plants	buildings grass sky
	person	person
	animal dog	animal dog

Рис. В.4





Image	Truth	Model pred
	person	person window
	grass road vehicle	road vehicle
	lake ocean water	beach lake ocean water
	flowers plants	flowers

Рис. В.5




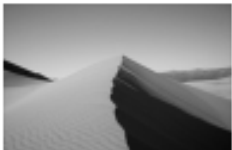
Image	Truth	Model pred
	snow	snow
	animal cat	animal cat
	animal elk lake water	animal lake water
	road sand	sand sky

Рис. В.6





Image	Truth	Model pred
	plane clouds	plane sky
	flowers garden grass mountain plants sky	clouds grass plants road sky
	animal beach	beach
	grass house	grass plants sky tree

Рис. В.7





Image	Truth	Model pred
	flowers plants	flowers plants
	animal	animal
	buildings clouds sky snow tower	buildings clouds sky
	animal bear snow	animal

Рис. В.8







Image	Truth	Model pred
	buildings	buildings town water
	clouds sky sunset	sky sunset
	person	person
	sky	clouds sky

Рис. В.9





Image	Truth	Model pred
	sky	grass sky
	road train window	sky train
	animal grass	animal grass
	water	person

Рис. В.10





Image	Truth	Model pred
	animal whales	animal fish
	beach clouds sky	clouds sky
	buildings	window
	grass house sky	buildings grass house

Рис. В.11



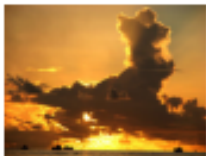
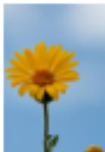
Image	Truth	Model pred
	beach clouds lake sky sunset beach water	clouds ocean sky sunset water clouds
	clouds ocean sky sunset water	lake ocean sky sunset water
	clouds sky sunset	clouds fire sky sunset
	flowers sky	flowers sky

Рис. В.12

Image	Truth	Model pred
	animal	animal
	person	person
	airport grass plane	airport clouds plane sky
	buildings clouds tower	sky tower

Рис. В.13