МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КОМП'ЮТЕРНОЇ ФІЗИКИ ТА ЕНЕРГЕТИКИ
Кафедра комп`ютерної фізики

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи
за освітньо-кваліфікаційним рівнем магістр

«Ефективне виявлення дублікатів зображень на основі аналізу контенту зображень»

Виконала: студентка 6 курсу, групи НФ-64
Скорнецька К.С.
спеціальність:
105 Прикладна фізика та наноматеріали
освітня програма: комп`ютерна фізика
Керівник: к.т.н., доц. Лісін Д.О.

Рецензент:

Харків – 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н.Каразіна

Навчально-науковий інститут комп'ютерної фізики та енергетики
Кафедра комп'ютерної фізики
Рівень вищої освіти магістр
Спеціальність 105 Прикладна фізика та наноматеріали
Освітня програма Комп'ютерна фізика

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ _____
підпис ініціали, прізвище

"____" _____20___року

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ (ПРОЕКТ)

_____ Скорнецька Катерина Сергіївна____
(прізвище, ім'я, по батькові студента)

1. Тема роботи __ Ефективне виявлення дублікатів зображень на основі аналізу контенту зображень___

керівник роботи____Лісін Денис Олександрович, к.т.н., доц._____.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від "10" лютого 2021 року №0210-05/374

2. Строк подання студентом роботи_____

3. Перелік питань, які потрібно розробити:

1) Проаналізувати сучасний стан наукового напряму щодо виявлення дублікатів зображень;

2) Визначити підпростір, охоплений дублікатами зображень:

3) Створити набір бінарних детекторів;

4) Розробити систему виявлення дублікатів зображень, на основі контенту.

4. План роботи

№ з/п	Назви етапів роботи
1.	Аналіз літератури по темі виявлення дублікатів зображень
2.	Написання вступу до роботи та першого розділу
3.	Написання другого розділу
4.	Написання третього розділу
5.	Формулювання висновків данної роботи

5. Дата видачі завдання « »_____2021 р

Студент _____ ___Скорнецька К.С.
підпис ініціали, прізвище

Керівник роботи _____ ____Лісін Д.О._____
підпис ініціали, прізвище

РЕФЕРАТ

Скорнецька Катерина. Ефективне виявлення дублікатів зображень на основі аналізу контенту зображень.

Дипломна робота представлена на 122 сторінках; складається з вступу, 3 розділів та висновків та список використаної літератури з 82 найменувань.

Метою даної роботи є розробка системи виявлення дублікатів на основі вмісту, на основі перевірки набору оригінальних зображень за допомогою розроблених бінарних детекторів дублікатів.

Об’єктом даної роботи є набори оригінальних зображень, з колекцій CGFA та MM270k.

Предмет дослідження — система виявлення дублікатів зображень на основі контенту зображень.

Розроблена загальна система виявлення дублікатів зображень на основі аналізу контенту зображень, за допомогою розроблених бінарних детекторів дублікатів.

КЛЮЧОВІ СЛОВА: ВИЯВЛЕННЯ ПОРУШЕННЯ АВТОРСЬКИХ ПРАВ, ВИЯВЛЕННЯ НЕЗАКОННИХ ЗОБРАЖЕНЬ, СИСТЕМА ВИЯВЛЕННЯ ДУБЛІКАТІВ, АНАЛІЗ ЗОБРАЖЕНЬ, МАШИННЕ НАВЧАННЯ, БАГАТОВИМІРНА ІНДЕКСАЦІЯ.

РЕФЕРАТ

Скорнецкая Екатерина. Эффективное обнаружение дубликатов изображений на основе анализа контента изображений.

Дипломная работа представлена на 122 страницах; состоит из вступления, 3 глав и выводов и список использованной литературы из 82 наименований.

Целью данной работы является разработка системы выявления дубликатов на основе содержания, на основе проверки набора оригинальных изображений с помощью разработанного бинарного детектора дубликатов.

Объектом данной работы является наборы оригинальных изображений, из коллекций CGFA и MM270k.

Предмет исследования - система обнаружения дубликатов изображений на основе контента изображений.

Разработана общая система обнаружения дубликатов изображений на основе анализа контента изображений, с помощью разработанных бинарных детекторов дубликатов.

КЛЮЧЕВЫЕ СЛОВА: ВЫЯВЛЕНИЕ НАРУШЕНИЯ АВТОРСКИХ ПРАВ, ВЫЯВЛЕНИЕ НЕЗАКОННЫХ ИЗОБРАЖЕНИЙ, СИСТЕМА ОБНАРУЖЕНИЯ ДУБЛИКАТОВ, АНАЛИЗ ИЗОБРАЖЕНИЙ, МАШИННОЕ ОБУЧЕНИЕ, МНОГОМЕРНАЯ ИНДЕКСАЦИЯ.

ABSTRACT

Skornetska Kateryna. Effective detection of duplicate images based on image content analysis.

Thesis is presented on 122 pages; consists of an introduction, 3 sections and conclusions and a list of references from 82 items.

The purpose of this work is to develop a system for detecting duplicates based on content by checking a set of original images using a developed binary duplicate detector.

The object of this work is a set of original images from the collections of CGFA and MM270k.

The subject of research - a system for detecting duplicate images based on image content.

The general duplicates of images detection system on the basis of the analysis of content of images is developed by means of the developed binary detector of duplicates.

KEY WORDS: DETECTION OF COPYRIGHT INFRINGEMENT, DETECTION OF ILLEGAL IMAGES, DUPLICATE DETECTION SYSTEM, IMAGE ANALYSIS, IMAGE ANALYSIS.

ЗМІСТ

ВСТУП 12
1 ЗАГАЛЬНИЙ МАТЕРІАЛ 16
1.1 Опис візуальної інформації 16
1.2 Індексація мультимедійних баз даних 21
1.3 Класифікація 24
1.4 Підсумок глави 30
2 СУЧАСНІ МЕТОДИ ВИЯВЛЕННЯ ДУБЛЬОВАНИХ ЗОБРАЖЕНЬ 32
2.1 Що таке виявлення дублікатів? 32
2.2 Фреймворк для виявлення копій зображень на основі вмісту 49
3 ЄДИНА ОРИГІНАЛЬНА СИСТЕМА ВИЯВЛЕННЯ ДУБЛІКАТІВ 76
3.1 Огляд системи 76
3.2 Система виявлення копій кількох оригінальних зображень 93
ВИСНОВКИ 112
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 115

ПЕРЕЛІК СКОРОЧЕНЬ ТА АБРЕВІАТУР

Математичні символи

f вектори;

I матриці;

f (·) функції;

S набори;

|R поля;

Pr {a > b} ймовірність;

Функція індикатора $I_A(x)$: дорівнює одиниці, якщо x ∈ A, і нулю в іншому випадку;

R(f) очікуваний ризик, пов'язаний з функцією класифікації f;

$R_{emp}(f)$ емпіричний ризик, пов’язаний із функцією класифікації f (і приклади);

w розділення гіперплощини;

y_i і-й ярлик;

x_i і-й приклад;

b запас;

ξ_i і-та змінна величина;

ν параметр компромісу у ν методі опорних методів;

α_i і-й множник Лагранжа у подвійній формі рішення мінімізації методу опорних векторів;

f (**z**) функція прийняття рішення для тестового зразку z;

ker ($\mathbf{x_i}$, $\mathbf{x_j}$) функція ядра між шаблонами $x_{i,j}$;

γ обернена ширина ядра для радіальної базисної функції;

σ ширина ядра для радіальної базової функції (тобто 1 / γ);

D(**I**) набір дублікатів зображення **I** (без композиції);

$\mathcal{E}_n$(**I**) набір дублікатів зображення **I** (композиція з n перетворень);

g_n(**I, p**) функціонал, що містить n послідовних операцій;

Ƒ(**I**, n) набір дублікатів зображення **I** (з використанням функціоналів g (·));

ν(**I**, ν) набір дублікатів зображення **I** (що дозволяє перестановку порядку перетворень);

T-тестове зображення;

u поріг ймовірності;

N кількість оригінальних зображень;

d_O^1 (**T**, u) функція детектора повторень для окремого оригінального зображення O;

d_O^N (**T**, u) функція детектора повторень для N оригінальних зображень, що містяться в O;

Ȯ набір оригінальних зображень;

T набір тестових зображень;

Ƒ набір не пов’язаних між собою тестових зображень;

L набір міток (-1 для непов'язаних зображень, +1, ..., N);

c_t істинний клас тестового зображення;

c_e оціночний клас тестового зображення;

c (c_t, c_e) функція індикатора помилок;

fp (c_t, c_e) функція хибнопозитивних показників;

fn (c_t, c_e) функція помилково негативного показника;

p_{FP}, $\hat{p}_{FP}$ реальна та розрахункова ймовірність помилково позитивних спрацьовувань;

p_{FN}, $\hat{p}_{FN}$ реальна та розрахункова ймовірність помилково негативних спрацьовувань;

C набір кандидатів, відібраних попереднім класифікатором ($C \subseteq Ȯ$);

R, G, B червоний, зелений та синій канали зображення;

H, S, I відтінок, насиченість та інтенсивність каналів зображення;

T P кількість справжніх спрацьовувань;

F P кількість помилкових спрацьовувань;

P кількість спрацьовувань;

F (T P, F P, P) f-оцінка, обчислена з використанням числа справжніх спрацьовувань, помилкових спрацьовувань та результатів;

ρ співвідношення між позитивними і негативними спрацьовуваннями;

$F\rho$ ($\hat{p}_{FP}$, $\hat{p}_{FN}$) f-оцінка, обчислений з використанням оцінених ймовірностей помилок;

f вектор функції, що представляє зображення;

W матриця зменшення розмірності;

x шаблон завантаження (перетворена версія f) в алгоритм машинного навчання;

δ розмір коробки, який використовується при пошуку R-дерева.

Скорочення

У наведеному нижче списку подано скорочення, використані у цьому документі. Скорочення також визначаються, коли вони вперше з'являються в тексті.

SVC Класифікатор векторної підтримки

VC-розмірність розмірність Вапника Червоненкіс

LOO по окремим об'єктам

CV перехресна перевірка

RBF радіальна базова функція

HSI інтенсивність насичення відтінку

ROC робоча характеристика приймача

DET компроміс з помилкою виявлення

KP ключовий момент

DPF функція сприйняття відстані

DCT дискретне косинусне перетворення

MSE середньоквадратична помилка

PCA аналіз основних компонентів

ICA аналіз незалежних компонентів

MPEG експертна група з рухомих зображень

JPEG спільна група експертів із зображень

SVDD опис опорних векторних даних

StirMark StirMark тестова версія 3.1

Qamra база Qamra

FN хибнонегативний (помилкове прийняття)

FP хибнопозитивний (помилкове відкидання)

TP true positive, класифікатор вірно відніс об'єкт до розглянутого класу

TN true negative, класифікатор вірно стверджує, що об'єкт не належить до розглянутого класу

P число спостережень позитивного класу

N число спостережень негативного класу

CGFA CGFA - віртуальний музей [Ke et al., 2004] -

MM270k MM270k - колекція комерційних зображень [Ke et al., 2004] -

F-оцінка f-оцінка

KNN k найближчий сусід

NN найближчий сусід

ВСТУП

Порівняно недавня простота, з якою цифровий контент можна виробляти, обробляти та розповсюджувати, відкрила нову еру - цілком цифровий світ. На жаль, ця революція також загострила старі проблеми та створила нові. Наприклад, незаконне розповсюдження захищених авторським правом або незаконних матеріалів, сьогодні дуже легко здійснити. Інша проблема стосується управління безліччю наявних документів.

Конкретна проблема стосується копіювання - точного чи приблизного, законного чи незаконного - документів. Дійсно, багато документів зберігаються на декількох серверах, і часто різні версії співіснують. З різних причин виникає необхідність виявлення копій даного документа.

Основною причиною є просто зменшення клопоту щодо управління документами. Але існує багато вторинних, але важливих причин, про що свідчать наступні додатки: моніторинг - відстеження документа, що обертається в Інтернеті, серед іншого для збору роялті, збору статистичних даних, виявлення порушень авторських прав та виявлення незаконних матеріалів; кластеризація - перегрупування документів, які є дублікатами при запиті до бази даних або Інтернету; пошук версій - пошук потрібної версії документа серед бази даних або Інтернету. Як бачимо, виявлення дублікатів має безліч корисних програм, і потреба в ефективному виявленні дублікатів зростає із збільшенням кількості генерованих цифрових документів. Щоб дати уявлення, IDC* підрахував, що людство створило 161 млрд гігабайт цифрової інформації в 2006 році, тоді як Каліфорнійський університет підрахував, що в 2003 році було створено лише п’ять мільярдів гігабайт, додатково підрахувавши, скільки місця буде витрачено, якщо не електронна інформація, наприклад, як аналогові радіопередачі або друковані службові записки, були оцифровані. Кількість дубльованих документів серед такої суми даних, безумовно, приголомшує, наприклад, IDC припускає, що в середньому кожен цифровий файл повторюється тричі.

Хоча виявити точну копію порівняно легко, виявити дещо модифіковану копію або майже дублікат є набагато складнішим завданням. У цьому дипломі представлена система виявлення дублікатів зображень на основі їх вмісту. Основна ідея полягає у створенні адаптованого детектора дублікатів для кожного зображення, дублікати якого повинні бути виявлені. Більшість інших робіт з виявлення дублікатів на основі вмісту зосереджені на пошуку опису зображення, стійкого до певних перетворень. Замість того, щоб вивчати цю вже набагато досліджену територію, ми пропонуємо дослідити, як розрізнити дублікати зображення та не пов'язані між собою зображення, даючи, можливо, не надійні описи зображень. Точніше, замість того, щоб знайти зображення, яке відповідає нашим потребам, ми справді розробляємо детектор, який відповідає характеристиці даного оригінального зображення та його копій.

Запропонована система виявлення дублікатів розробляється у два етапи. На першому етапі створюється набір бінарних детекторів, кожен адаптований до певного оригінального зображення. Точніше, кожен детектор здатний розрізняти дублікати оригінальних та не пов'язаних між собою зображень. Вони складаються з трьох етапів, викладених далі. На першому кроці тестове зображення попередньо обробляється таким чином, щоб додати певний ступінь незмінності щодо загальних операцій обробки зображень. На другому етапі для опису зображення використовується глобальна статистика. Нарешті, на останньому кроці для визначення ймовірності того, що тестове зображення є дублікатом оригінального зображення, використовується нелінійна функція прийняття рішення, заснована на класифікаторі опорних векторів. На другому етапі двійкові детектори ефективно складаються разом, щоб сформувати кілька повторюваних детекторів оригінальних зображень. Іншими словами, система здатна визначити, тестове зображення є дублікатом одного з оригіналів чи не пов'язане з яким-небудь із них. Повна система складається з трьох етапів, наведених далі. На першому кроці найбільш вірогідні оригінали ефективно відбираються за допомогою адаптованої техніки індексування. Вони формують набір кандидатів. На другому етапі бінарні детектори, розроблені на першому

етапі, застосовуються до кожного елементу набору кандидатів. Нарешті, обирається оригінал, що відповідає найбільшій імовірності серед набору кандидатів. Система оцінює, що тестове зображення є дублікатом цього оригіналу, якщо відповідна ймовірність перевищує певний поріг.

Основною ідеєю запропонованої архітектури є адаптивність. Наприклад, він адаптує метрику виявлення до кожного оригінального зображення, відомого системі. Крім того, новий оригінал можна додати до системи без необхідності перекваліфікувати вже відомий оригінал. Більше того, систему можна легко адаптувати до нових дублікатів, якщо помічено, що вони уникнуть виявлення.

Значні внески роботи, представленої в цьому дипломі, узагальнені нижче.

- Сучасний рівень розпізнавання дублікатів на основі вмісту. Наскільки мені відомо, сучасний рівень даної дипломної роботи є першим вичерпним звітом про існуючі методики виявлення дублікатів на основі вмісту.
- Визначення підпростору, охопленого дублікатом зображення. Підпростір визначається у кілька етапів. На першому етапі розглядаються параметризовані перетворення зображення, і композиція не допускається. У цьому випадку кожне перетворення генерує криву в просторі зображень, а підпростір, охоплений дублікатами, відповідає об'єднанню цих кривих. На другому етапі вивчається ефект складених перетворень. У цьому випадку підпростір, охоплений дублікатами, являє собою багатоманітність, якщо порядок операцій не має значення. Розмірність різноманітності обмежена верхньою кількістю розглянутих перетворень.
- Визначення загальної системи виявлення дублікатів. Система виявлення дублікатів, представлена в цьому дипломі, організована навколо ідеї адаптивного виявлення. Іншими словами, система знає оригінальні зображення, для яких вона повинна виявляти дублікати, і може адаптуватися до характеристик кожного оригіналу. Точніше, запропонована система може вирішити, чи є вхідне зображення дублікатом одного з оригіналів, що міститься в його колекції, чи не пов’язане з яким-небудь із них. Порівняйте цей підхід із класичною парадигмою пошуку зображень, де зображення, подібні до запиту,

отримуються з бази даних. Оригінал може бути запитом або міститися в базах даних. Хоча підхід до отримання зображень є більш гнучким, він навряд чи пристосований до внутрішніх характеристик кожного оригіналу та його копій.
• Визначення та кількісна оцінка продуктивності адаптивних бінарних детекторів-дублікатів. Бінарний детектор - це дублікат детектора, адаптований до конкретного оригінального зображення. Це дає оцінку ймовірності того, що тестове зображення є дублікатом оригінального зображення. Він базується на низькорівневому візуальному описі зображення і побудований навколо класифікатора опорних векторів.

- Визначення та кількісна оцінка продуктивності дубліката попереднього класифікатора. Система виявлення дублікатів на основі бінарних детекторів перевіряє тестове зображення з кожним бінарним детектором, кожен з яких відповідає відомому оригінальному зображенню. Це стає громіздким із збільшенням кількості оригіналів. Щоб уникнути цього, попередній класифікатор дублікатів ефективно відбирає підмножину оригінальних зображень таким чином, що тестове зображення, швидше за все, буде дублікатом одного з них. Він заснований на низькорівневому візуальному описі зображення і побудований навколо просторового методу доступу.
- Кількісна оцінка продуктивності системи виявлення дублікатів. Ефективність, отримана всією системою, оцінюється на різних колекціях зображень. Встановлено, що запропонований підхід працює дуже добре у порівнянні з сучасними системами виявлення дублікатів на основі вмісту.

1 ЗАГАЛЬНИЙ МАТЕРІАЛ

У цьому розділі представлені деякі ключові концепції та алгоритми, необхідні для кращого розуміння решти глав. Текст організований у три взаємопов’язані частини, а саме візуальний опис інформації, індексація мультимедійних баз даних та машинне навчання. Опис візуальної інформації представлений у частині 1.1. Він дає огляд різних можливостей для опису зображення за допомогою низькорівневих функцій. Індексація мультимедійних баз даних представлена в частині 1.2. Він пропонує загальний огляд методів, що використовуються для ефективної організації та отримання об’єктів з баз даних, що індексують багатовимірні ознаки. Нарешті, машинне навчання представлено в частині 1.3. Це знайомить читача з областю машинного навчання, а точніше, з сферою контрольованої класифікації.

1.1 Опис візуальної інформації

Візуальні дескриптори дають статистичні дані про зображення. Хороший дескриптор дозволяє розрізняти схожі та несхожі зображення. Зауважте, що поняття подібності дуже залежить від застосування. Наприклад, подібність означає "візуально узгоджені зображення" в рамках пошуку зображень, тоді як означає "візуально майже ідентичні" при виявленні дублікатів.

Далі подано чотири типи дескрипторів зображень низького рівня. Перший тип дескрипторів, представлений у частині 1.1.1, стосується кольорового вмісту зображення. Другий тип дескрипторів, поданий у частині 1.1.2, стосується текстури, яка стосується структурованого візуального мотиву. Завдяки своїй простоті та порівняно низькій обчислювальній вартості, колір і текстура є двома найпоширенішими дескрипторами низького рівня при отриманні зображень. Третій тип дескрипторів, представлений у частині 1.1.3, стосується опису на основі області, який включає не тільки опис областей з використанням кольору або текстури, але також і опис форми області. Місцеві

описи зображення багатші та дискримінаційніші, ніж його загальний опис. Однак основним недоліком дескрипторів областей є необхідність сегментування для отримання значущих областей. Нарешті, четвертий тип дескрипторів, введений у частині 1.1.4, стосується відомих точок, які дозволяють отримувати локальні описи зображень, уникаючи сегментації.

1.1.1 Кольори

Кольорові дескриптори - це, можливо, найпоширеніші функції при отриманні зображень [1]. Основна причина полягає в тому, що кольорові дескриптори відносно надійні до фонових ускладнень і не залежать від розміру зображення.

Гістограма кольорів - найпоширеніший дескриптор кольору. Він дає кількісну оцінку розподілу ймовірностей інтенсивності кольорових каналів. Незважаючи на те, що гістограми легкі для обчислення та містять багато інформації, вони мають три важливі недоліки. По-перше, вони часто розсіяні і, отже, досить чутливі до шуму. По-друге, оскільки гістограми є квантованими версіями основних розподілів ймовірностей, порівняти дві гістограми не просто. По-третє, гістограми важко індексувати через їхню багатовимірність, див. Частина 1.2 для отримання додаткової інформації про багатовимірні методи доступу.

Щоб врахувати перцептивний вплив кольорів, набори кольорів використовуються в [2]. При такому підході кольоровий простір RGB спочатку трансформується у перцептивно більш рівномірний колірний простір, наприклад HSV. Згодом однорідний за сприйняттям кольоровий простір квантується в контейнери таким чином, що кожен контейнер відповідає кольору, який може однозначно позначити глядач. Цей підхід заснований на ідеї про існування невеликої кількості кольорів, які майже ніколи не змішуються [2].

Розроблені нещодавно, домінуючі кольори забезпечують компактний і простий в індексі дескриптор кольорів [3]. Більше того, це стандартний дескриптор у MPEG-7 [3]. Точніше, кольори на зображенні за допомогою векторної квантизації згруповані в невелику кількість репрезентативних або домінуючих кольорів. Дескриптор ознак складається з репрезентативних кольорів, їх відсотків, просторової когерентності та дисперсії кольорів. Однією з переваг домінуючого кольору є те, що його можна індексувати у тривимірному кольоровому просторі, що дозволяє уникнути проблем із індексуванням великих розмірів, пов’язаних із традиційною гістограмою кольорів. Тим не менше, недоліком, пов'язаним з домінуючим кольором, є етап векторної квантизації, який може бути відносно дорогим з точки зору обчислювальних ресурсів.

За винятком домінуючого дескриптора кольору, дескриптори кольору зазвичай не враховують просторовий розподіл кольорів. Було помічено, що системи пошуку зображень, засновані лише на кольоровій статистиці, мають тенденцію повертати занадто багато помилкових відповідей [4].

1.1.2 Текстура

Текстура відноситься до структурованого візуального мотиву. Це важлива складова будь-якого візуального об'єкта, такого як ліс, хмари чи гори. Характеристики текстури кількісно визначають випадкові, але структуровані зміни інтенсивності (або кольору). Точніше, ознаки вимірюють варіацію інтенсивності поверхні та кількісно визначають такі властивості, як гладкість та регулярність. Текстура, як і колір, є потужним дескриптором низького рівня для опису зображення. Текстури описують важливу інформацію про структурне розташування поверхонь, а також їх зв'язок із оточенням [1].

Статистичні методи характеризують текстури за статистичними властивостями рівнів сірого пікселів. Як правило, ці властивості обчислюються

на основі матриці спільного виникнення рівня сірого на поверхні [2]. Багато дослідників досліджували цей тип підходу, і експериментально було встановлено, що контраст, момент зворотного відліку та ентропія є трьома властивостями, які надають найкращу дискримінаційну силу [4].

Нарешті, недавній прогрес у характеристиці текстур стосується багатомасштабних підходів. Наприклад, деякі методи полягають у описі текстур як простої статистики хвильових перетворень (а саме середнього значення, дисперсії або перекосу) розподілу коефіцієнтів хвиль для кожного піддіапазону [3]. Існує безліч можливих хвильових перетворень. Однак було встановлено, що хвиля Габора дає найкращу дискримінаційну силу [5].

1.1.3 Область

Область - це візуально чи навіть семантично значуща частина зображення. Не тільки область може бути представлена кольором або текстурою (за допомогою дескрипторів, представлених у частині 1.1.1 або частині 1.1.2), але і вона може бути описана за формою. Опис фігури можна розділити на дві категорії, а саме на межі та області. Перша категорія використовує лише контур регіону, тоді як друга категорія використовує всю область [1]. У будь-якому випадку для опису області потрібно визначити межу області. Для цього потрібна або ручна, або автоматична сегментація зображення.

Багато дескрипторів фігури на основі межі базуються на дескрипторі Фур'є [6]. Іншими словами, перетворення Фур'є межі використовується як ознака фігури. Пізніше деякі дослідники вдосконалили дескриптори Фур'є, додавши шуму стійкість, а також незмінність геометричного перетворення [1].

Багато дескрипторів фігур на основі області використовують інваріанти моменту. Сім моментів, інваріантних до перетворень, було вперше

запропоновано в [6]. У більшості наступних робіт використовуються варіанти цих семи інваріантів моменту.

1.1.4 Виділена точка

Опис виділених точок - це можливий спосіб запропонувати місцеві описи, уникаючи сегментування зображення. Коротше кажучи, методи виділених точок концентрують локальний опис у декількох векторах ознак, кожен з яких відповідає фіксованій області навколо виділеної точки. Виділені точки - це не що інше, як конкретні пікселі зображення, описи яких є найбільш помітними (з урахуванням деяких критеріїв) серед усіх пікселів зображення.

Оскільки опис зображення зведено до обмеженої кількості векторів особливостей, виділені точки слід вибирати, щоб він мав велику помітність і доведену стійкість [6]. Однією з ранніх і дуже популярних робіт з виявлення помітних точок є робота [7], де виявляються кути об'єктів. У цьому випадку для вибору виділених точок використовується поняття кутів та країв, а не міра стійкості. Це призводить до моментів, які можуть бути не дуже надійними для перетворень зображень. З цієї причини виділення часто визначають як точки, які найдовше виживають до певних перетворень, наприклад, для поступового розмиття зображення [7].

В даний час найбільш успішним відомим дескриптором точок є той, що представлений у [5]. У цьому підході виділені точки є локальними екстремумами в масштабному просторі подання зображення (отриманого через серію розмиття Гауса зображення). Потім кожна точка інваріантно описує орієнтацію країв, що містяться в області, що оточує виділену точку. Загалом, метод Лоу описує типове зображення з використанням кількох сотень яскравих точок, але для складних сцен може знадобитися кілька тисяч точок. Тим не менше, хоча дескриптор Лоу досягає хороших показників, це обчислювально дорого. З цієї причини деякі дослідники розробили методи зменшення його

обчислювальних витрат. Наприклад, [6] зменшує кількість точок, необхідних для опису зображення, одночасно досягаючи кращих результатів відповідності. Нарешті, модифікація Grabner et al. досягає прискорення порядку восьми до десяти щодо оригіналу Лоу шляхом апроксимації розмиття Гауса [7].

1.2 Індексація мультимедійних баз даних

Залежно від програми, мультимедійні бази даних потребують різних властивостей і повинні підтримувати різні типи запитів. Запит на пошук або метод доступу до мультимедійної бази даних часто вимагає швидкого виконання геометричної операції пошуку, наприклад, запиту точки або регіону. Обидві операції вимагають швидкого доступу до тих об'єктів даних у базі даних, які займають певне місце в просторі.

Крім того, мультимедійні об'єкти часто живуть у просторі, що містить багато вимірів. Існує багато опитувань щодо багатовимірних методів індексації, що використовуються для мультимедійних баз даних. Для отримання додаткової інформації читача посилаються на [8]. Мультимедійні бази даних мають важливе значення в багатьох областях застосування, таких як географія, САПР, медицина чи пошук зображень.

1.2.1 Багатовимірний доступ

Як було видно раніше, для підтримки операцій пошуку, необхідних мультимедійним базам даних, потрібні спеціальні багатовимірні методи доступу. Однак головна проблема в розробці таких методів полягає в тому, що серед просторових об'єктів не існує повної впорядкованості, щоб зберегти просторову близькість. Іншими словами, немає відображення з двовимірного або багатовимірного простору в одновимірний простір, щоб будь-які два

просторово близькі об'єкти у багатовимірному просторі також були близькі один до одного в одновимірній відсортованій послідовності [8].

З цієї причини проектування ефективних методів доступу в багатовимірних просторах набагато складніше, ніж у традиційних базах даних, де доступно багато ефективних методів доступу. Приклади таких одновимірних методів доступу (також звані структурами одного ключа) включають B-tree [3] та розширюване хешування [8]. Популярний підхід до обробки багатовимірних пошукових запитів полягає у використанні однієї структури ключів для кожного виміру. На жаль, такий підхід може бути дуже неефективним, оскільки кожен індекс проходить незалежно від інших. Отже, вибірковість в одному вимірі не може бути використана для звуження пошуку в інших вимірах [9]. Загалом, немає простого і очевидного способу розширення окремих ключових структур для обробки багатовимірних даних [7].

Іншими словами, мультимедійні бази даних вимагають реальних багатовимірних методів індексації. Перш ніж продовжувати далі, зауважте, що при збільшенні розмірності простору даних можна спостерігати кілька математичних ефектів. Часто ці ефекти неможливо інтуїтивно обгрунтувати, просто поширивши дво- чи тривимірний досвід на простори великих розмірів [8]. Деякі ефекти представляють лише математичний інтерес, тоді як інші мають важливе значення для продуктивності багатовимірних індексних структур. Тому у світі баз даних ці ефекти узагальнюються під парасолькою «прокляття розмірності [4]». Якісно кажучи, такі важливі параметри, як обсяг і площа, експоненційно залежать від кількості розмірів простору даних. Отже, багато традиційних структур індексації працюють ефективно лише в тому випадку, якщо кількість розмірів досить мала [3]. Це означає, що більшість із них непридатні для індексації мультимедійних баз даних.

Багатовимірні методи доступу до даних можна класифікувати на дві категорії, а саме точкові та просторові методи доступу [8]. Методи точкового доступу в першу чергу призначені для здійснення просторового пошуку в базах даних точок, в яких зберігаються лише багатовимірні точки (без просторового

розширення). З іншого боку, просторові методи доступу керують об’єктами, які мають просторові характеристики на додаток до своїх позицій у просторі. Наприклад, такими об’єктами є лінії, багатокутники або багатовимірні багатогранники.

1.2.2 Методи точкового доступу

Як правило, методи доступу до точок упорядковують точкові дані у сегменти, кожна відповідає якомусь підпростору Всесвіту. Деякі методи доступу до точок використовують одновимірне хешування для індексування двовимірних точок. Хоча немає повного впорядкування d-вимірних об'єктів в одному вимірі, ці методи використовують евристичні прийоми, щоб забезпечити однакове індексування двох об'єктів, близьких один до одного в багатовимірному просторі. Інші методи точкового доступу використовують ієрархічні структури даних для управління точковими даними. Нарешті, такі методи доступу, як дерево Бадді [9], є гібридними, оскільки вони включають як ієрархічні, так і хеш-методи.

1.2.3 Методи просторового доступу

Точкові методи доступу не можуть безпосередньо управляти об’єктами просторово, але їх часто розширюють, щоб покрити цю потребу. Геде і Гюнтер класифікують методи точкового доступу відповідно до технік, що використовуються для розширення від точкових до просторових методів доступу. Найважливіші підходи до розширення викладені далі.

Методи відображення об’єктів відображають геометричні об’єкти в точки в просторовому просторі. Наприклад, прямокутник у R^2 відповідає точці в R^4. Згодом для управління точками використовуються існуючі методи доступу до точок.

Методи обмеження об’єктів є найпопулярнішими методами просторового доступу. У цих підходах простір розкладається ієрархічно. Об'єкти зберігаються в листках ієрархічних структур, а проміжні вузли використовуються для виконання ефективних запитів. Оскільки просторове розширення вузлів на одному рівні може перекривати один одного, кількість шляхів, якими слід дотримуватися під час запиту, варіюється. Найбільш перспективними методами обмеження об'єктів є R-дерево [6] та R* -дерево [9].

Як і методи обмеження об'єктів, методи відсікання використовують ієрархічні структури даних. Однак відсікання використовується для запобігання перекриванню проміжних вузлів на одному рівні. Точніше, об’єкти обрізаються або підрозділяються і зберігаються в декількох вузлах, щоб гарантувати проміжні вузли, що не перекриваються. Таким чином, під час запиту пройдено лише один шлях ієрархічної структури [7].

1.3 Класифікація

Класифікація полягає у пов’язуванні об’єкта з одним із декількох класів відповідно до значущого правила класифікації. Можлива формалізація цієї проблеми полягає в наступному. Кожен клас представлений міткою $y_i \in L \subset Z$ об’єкт моделюється вектором $x \in R^d$, позначеним шаблоном, а правило класифікації задається функцією $f: R^d \rightarrow L$, що відображає будь-який d-мірний вектор x на а ярлик y. Машинне навчання забезпечує безліч автоматичних методів для ефективного проектування класифікаційних функцій на основі прикладів. Алгоритми навчання можна диференціювати за обсягом інформації, яка їм надається, а саме контрольовані методи мають приклади у вигляді пар (x_i, y_i), тоді як некеровані алгоритми мають доступ лише до векторів x_i.

Далі нас в основному цікавлять керовані алгоритми. Презентація складається з двох частин. Перша частина вводить поняття машинного навчання, що не залежать від алгоритму, в той час як друга частина дає огляд

конкретного, але потужного методу класифікації, а саме класифікатора опорних векторів. За винятком частинау 2.3.3, розглядається лише двокласний випадок; іншими словами, | L | = 2.

1.3.1 Огляд статистичного машинного навчання

Від очікуваного до емпіричного ризику

Хороший класифікатор - це в основному той, який добре узагальнює невидимі (або нові) моделі. Іншими словами, хороший класифікатор повинен з великою ймовірністю зіставити нові шаблони з правильними мітками. Це поняття узагальнення можна формалізувати шляхом введення функції втрат l: L × L ⟶ R, яка характеризує вартість відображення шаблону до неправильного класу. Якщо припустити, що пари (x, y) нових моделей, а мітки самостійно витягуються з невідомого розподілу ймовірностей p (x, y), очікуваного ризику R, пов'язаний з функцією класифікації f

$$R(f) = \int 1\big(y, f(x)\big) dp(x, y). \qquad (1.1)$$

Теоретично достатньо вибрати функцію f серед набору доступних класифікаційних функцій F, яка мінімізує очікуваний ризик для отримання найкращого класифікатора. На практиці, однак, виникає кілька ускладнень, і необхідно проводити наближення. Наприклад, використання функції втрат l (y, ŷ) = $I_{\{y\}}$ (ŷ) дає очікувану середню кількість класифікаційних помилок. Хоча теоретично приваблива, ця функція втрат призводить до нерозв'язних проблем оптимізації. Для отримання практично здійсненних алгоритмів функція індикатора зазвичай апроксимується гладкими функціями, які нижньо та верхньо обмежені 0 та 1 відповідно. Подальше наближення стосується очікуваної вартості, яку практично неможливо розрахувати, оскільки основний розподіл ймовірностей p невідомий. Для вирішення цієї проблеми часто використовується емпіричний ризик (середня вартість навчального набору)

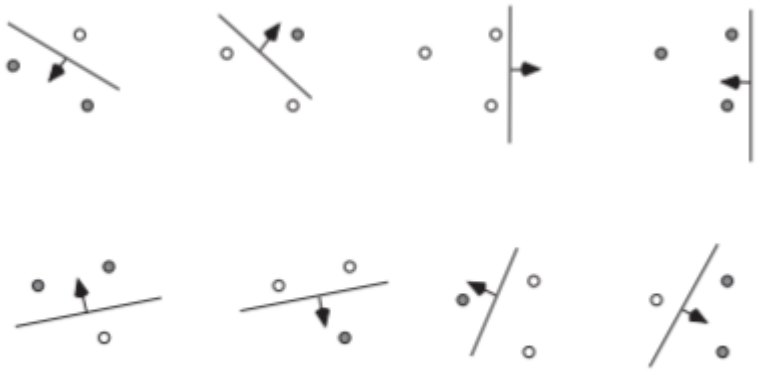


Рисунок 1.1 Три точки в R^2 розбиті орієнтованими напівплощинами. Вісім можливих призначень міток можна правильно класифікувати, використовуючи різноорієнтовані напівплощини, що означає, що VC-розмір набору орієнтованих напівплощин становить принаймні три (насправді рівно три, оскільки розбиття робить неможливим додавання четвертого пункту). Загалом, VC-розмірність напівплощин у R^n дорівнює n + 1. Крім того, зверніть увагу, що n + 1 вирівняні точки в R^n не можуть бути зруйновані орієнтованими напівплощинами. Ця цифра надана Берджесом [8].

замість очікуваного ризику

$$R_{emp}(f) = \frac{1}{M} \sum_{i=1}^{M} l(y_i, f(x_i)) \quad (1.2)$$

У цьому випадку функцію класифікації f можна знайти наступним чином

$$f = \arg\min_{g \in \mathbb{F}} R_{emp}(g) \quad (1.3)$$

де x_i - M відомих прикладів, а y_i - відповідні мітки. Цікавим питанням є те, чи мінімізація емпіричного ризику призводить до мінімального очікуваного ризику. Відповідь є ствердною, лише якщо розмір навчального набору має тенденцію до нескінченності. Однак, якщо кількість навчальних прикладів обмежена, мінімізований емпіричний ризик може стати меншим ніж мінімальний очікуваний ризик. Отже, мінімізація емпіричного ризику може

бути неоптимальною, і класифікатор може не узагальнювати так добре, як очікувалося.

Ємність та її зв'язок із кількістю прикладів навчання

Для більшості програм кількість навчальних прикладів обмежена, і, отже, мінімізація емпіричного ризику, ймовірно, буде неоптимальною. Мінімальна кількість навчальних прикладів, необхідна для пошуку класифікаційної функції f, яка добре працює, зазвичай залежить від ємності набору функцій F. Потенціал вимірює внутрішню складність набору функцій. Конкретним прикладом вимірювання ємності є розмірність Вапніка Червоненкіса (VC-розмірність), яка кількісно визначає максимальну кількість точок, які можуть бути зруйновані набором функцій. Точніше, якщо для даного набору h точок кожне з 2^h можливих призначень міток правильно позначене функцією F, тоді h точки розбиваються F. Для ілюстрації цієї точки, давайте розглянемо три точки та набір орієнтовані напівплощини, зображені на рисунку 1.1. Хоча теоретично цікавий VC-вимір не є практичним, оскільки його складно обчислити в більшості випадків. Тим не менше, це приносить важливий «погляд назад». Точніше, VC-розмірність набору функцій F дозволяє обмежувати мінімальний очікуваний ризик на

$$\min_{g \in \mathbb{F}} R\,(g) \leq \min_{g \in \mathbb{F}} R_{emp}(g') + \sqrt{\frac{h\,(\ln\frac{2M}{h}+1)-\ln\frac{\delta}{4}}{M}} \qquad (1.4)$$

де нерівність виконується з імовірністю більшою за 1 - δ при M > h. Якщо кількість прикладів M прагне до нескінченності, межа стає жорсткішою, оскільки другий доданок у правій частині рівняння прагне до нуля. У цьому випадку функція f, яка мінімізує емпіричний ризик, також мінімізує очікуваний ризик. Однак для фіксованої кількості прикладів єдиним способом отримання більш жорсткої межі є зменшення VC-розмірності h. Це спостереження призводить до наступного емпіричного правила. Якщо кількість навчальних прикладів невелика, потужність набору функцій також повинна бути невеликою. І навпаки, якщо кількість прикладів зростає, слід отримувати кращі

класифікатори, використовуючи набори функцій з більшими можливостями. Це правило відповідає інтуїтивно зрозумілому принципу бритви Оккама: «За всіх рівних умов найпростіше пояснення є найкращим.»

Регуляризація як засіб контролю потужності

У випадку обмеженої кількості прикладів можливий спосіб отримання хороших класифікаторів полягає у наявності параметричного набору функцій F (λ), де λ контролює свою ємність. Хоча пряме визначення такого набору не є тривіальним, його можна легко побудувати побічно, використовуючи підхід, який називається регуляризацією. Дійсно, регуляризована емпірична мінімізація ризику звучить наступним чином

$$f_{\lambda} = \arg\min_{g \in F(\lambda)} R_{emp}(g) = \arg\min_{g \in F(\lambda)} \sum_{i=1}^{M} 1\big(y_i, g(x_i)\big) + \lambda \cdot \Omega(g) \quad (1.5)$$

де функціонал регуляризації Ω: F → R приймає значення, пропорційні складності функцій g, а λ - невід'ємне дійсне. Як результат, λ дозволяє ефективно управляти ємністю F, оскільки чим більший λ, тим менша ємність відповідного набору функцій. На практиці функціоналом регуляризації часто є L_2-норма, якщо функції f є елементами простору Гільберта, у цьому випадку F (λ) все більше і більше зміщується до нормальних функцій із збільшенням λ. Хоча емпіричний ризик зазвичай зменшується із збільшенням потужності, очікуваний ризик спочатку зменшується, щоб досягти мінімуму, перш ніж знову збільшуватися. Малюнок 1.2 ілюструє це явище, зауважте, що зона праворуч від оптимальної ємності відповідає переобладнаним класифікаторам, тоді як ліворуч відповідає недостатньо обладнаним класифікаторам. Потрібно знайти потенціал, що спричиняє оптимальний очікуваний ризик, що передбачає оцінку очікуваного ризику.

1.3.2 Оцінка очікуваного ризику

Більшість класифікаторів мають вільні параметри, які потрібно налаштувати, щоб отримати хороші результати, часто опосередковано контролюючи пропускну здатність основного набору функцій. Як зазначено у частині 1.3.1, оптимальний вибір цих параметрів відповідає мінімальному очікуваному ризику. Це означає, що очікуваний ризик повинен бути оцінений. Процедура перехресної перевірки є популярною методикою оцінки очікуваного ризику для алгоритмів довільної класифікації. В k-кратному алгоритмі перехресної перевірки шаблон навчання випадковим чином розбивається на k взаємовиключних підмножин приблизно однакового розміру. Функція класифікації отримується шляхом навчання на k - 1 підмножин, а потім перевіряється на решті підмножини. Ця процедура повторюється k разів, причому кожна підмножина використовується для тестування один раз. В середньому похибка тесту за k випробувань дає оцінку очікуваного ризику. Показано, що цей метод дає хорошу оцінку помилки узагальнення [Duan et al., 2003]. З іншого боку, це передбачає багато обчислень, оскільки класифікатор потрібно навчити k + 1 рази замість одного разу. Одноразова оцінка є крайнім випадком методу перехресної перевірки. Одноразова оцінка полягає у використанні стількох згибів, скільки є прикладів навчання.. Хоча обчислювально дорого, відомо, що одноразова оцінка майже неупереджена. Є безліч постійних досліджень щодо того, як ефективно обмежити оцінку, яка не стосується одного разу. Більша частина обмежень залежить, однак, від використовуваної методики класифікації.

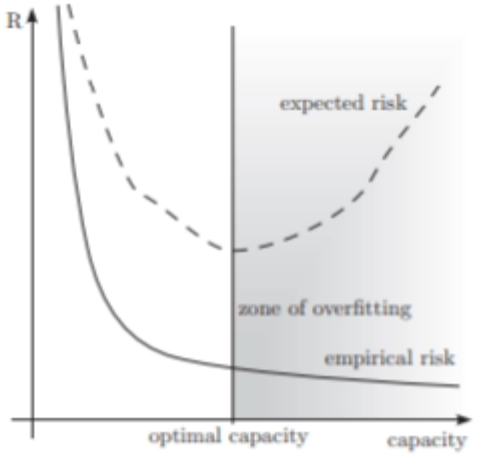


Рисунок 1.2 Емпіричні та очікувані ризики для різних можливостей. Емпіричний ризик зменшується із збільшенням потужності, тоді як очікуваний ризик досягає мінімуму, перш ніж знову збільшуватись. Зона правої руки оптимальної ємності відповідає переобладнаним класифікатором.

1.4 Підсумок глави

Цей частина стосується загальних матеріалів, які допомагають краще зрозуміти решту дисертації. Він складається з трьох не пов’язаних між собою частин: візуального опису, індексації мультимедійних баз даних та машинного навчання.

Частина візуального опису спочатку пояснює ідею опису зображень через функції низького рівня. Потім він переходить до представлення різних існуючих типів дескрипторів низького рівня. Ці типи - це колір, фактура, область та помітні точки. Слід зазначити, що колір та текстура є найбільш часто використовуваними дескрипторами у програмах пошуку зображень. Крім того, зазначається, що дескриптори регіонів потребують сегментації зображення, і з цієї причини вони менш цікаві, хоча вони надають місцеві описи зображень. Нарешті, основні моменти представлені як методи, що забезпечують локальний опис, уникаючи підводних каменів сегментації.

Частина мультимедійних баз даних спочатку описує різницю між традиційними та мультимедійними базами даних. Виявляється, бази даних, що містять візуальні функції, вимагають багатовимірних методів доступу. Ця вимога означає, що звичайні методи індексації, наприклад, засновані на хешуванні, безпосередньо не використовуються. Потім представлені два типи багатовимірних методів доступу. Перший тип методів, який називається методами точкового доступу, використовується для індексації багатовимірних точок, тоді як другий тип методів, називається просторовими методами доступу, використовується для індексації багатовимірних точок, які додатково мають просторове розширення.

Нарешті, частина машинного навчання спочатку визначає проблему класифікації. Потім текст дає огляд статистичного машинного навчання. Виявляється, методи класифікації оцінюються з використанням очікуваного ризику або кількості помилок, допущених при класифікації нових моделей. Для того, щоб вибрати хороший класифікатор, очікуваний ризик, таким чином, потрібно оцінити, наприклад, використовуючи методи перехресної перевірки.

2 СУЧАСНІ МЕТОДИ ВИЯВЛЕННЯ ДУБЛЬОВАНИХ ЗОБРАЖЕНЬ

Проблема виявлення дублікатів зображень виникає з різних - і часто не пов'язаних між собою - ділянок. В результаті існують різні визначення та рішення проблем. Спочатку ми вільно (непевно) визначаємо проблеми виявлення дублікатів у частині 2.1. Потім у частині 2.2 порівнюються два досить різнорідні рішення, а саме водяні знаки та виявлення дублікатів на основі вмісту. Нарешті, у частині 2.3 представлені та проаналізовані існуючі на основі вмісту методи виявлення дублікатів.

2.1.1 Що таке виявлення дублікатів?

Наведено визначення виявлення дублікатів. Виявлення дублікатів - це завдання, спрямоване на виявлення дублікатів оригінального зображення. Отже, спочатку необхідно визначити, що таке дублікат. Коротше кажучи, дублікат - це трансформована версія оригінального твору мистецтва, що зберігає подібну візуальну цінність. Іншими словами, «бути дублікатом» - це парне відношення еквівалентності, яке пов’язує оригінал з будь-якими його варіаціями за допомогою операції перетворення, наприклад, стиснення, зміни яскравості або обрізання. За розширенням, якщо зображення A є дублікатом іншого зображення B, а ще одне зображення C є дублікатом зображення B, то зображення C, в свою чергу, є дублікатом зображення A.

Нарешті, завдання виявлення дублікатів може бути виражене наступним чином. Виявлення дублікатів спрямоване на виявлення всіх дублікатів певного зображення серед колекції зображень. Або в спрощеній формі мета виявлення дублікатів полягає в тому, щоб визначити, чи є два подані зображення дублікатами один одного чи не пов’язані між собою. Це наївне визначення, яке контекстуалізує цей стан техніки, тоді як більш офіційне - у частині 3.

2.1.2 Дві філософії виявлення дублікатів

Існують дві дуже несхожі філософії виявлення дублікатів, а саме методи, засновані на водяних знаках, та підходи, що базуються на вмісті. Підхід водяних знаків полягає у вбудовуванні підпису в оригінальне зображення перед розповсюдженням роботи. Потім можна виявити копії оригіналу, перевіривши наявність підпису на зображеннях. З іншого боку, підхід, заснований на вмісті, спирається, як пропонується його назва, на аналізі вмісту зображення з метою вилучення відповідних візуальних особливостей. Потім копії ідентифікуються, коли їх характеристики близькі до характеристик оригінального зображення. У наступних двох підчастинаах ці дві філософії представлені більш докладно та проаналізовано їх переваги та недоліки

Виявлення дублікатів на основі водяних знаків

Історично виявлення дублікатів зображення в основному виконувалось із використанням методів водяних знаків. Ідея водяних знаків досить проста: власник авторських прав на вміст надійно та непомітно включає секретний підпис у зображення до його розповсюдження. Прихований підпис служить двом цілям. По-перше, це дозволяє ідентифікувати власника вмісту у судових справах. По-друге, це дозволяє виявляти копії вмісту, наприклад, переглядаючи Інтернет, і згодом визначати, чи копія використовується законно чи незаконно. Існує багато книг та опитувань щодо водяних знаків, оскільки ця сфера обробки сигналів є більш зрілою [наприклад, Barnett, 1999; Кокс та ін., 2001; Кокс і Міллер, 2002; Hartung and Kutter, 1999].

Нещодавно водяні знаки, як засіб захисту вмісту, зазнали жорсткої критики. Херлі розпочав дискусію щодо недоліків водяних знаків суперечливою роботою під назвою "Чому водяні знаки - це нонсенс [9]". Суть аргументації Герлі полягає в тому, що захист усіх об’єктів у невеликому районі позначеного об’єкта, як це виконується в більшості опублікованих алгоритмів водяних знаків, є необхідним, але недостатнім. Він продовжує, стверджуючи,

що корисний алгоритм водяних знаків повинен захищати всі цінні варіації, а не лише ті, що знаходяться поблизу від позначеного об'єкта. Однак Мулин частково відкинув твердження про те, що "водяні знаки - це нісенітниця", зауваживши, що може бути досить важко свідомо знайти важливу трансформацію зазначеного об'єкта, яка уникає виявлення. Крім того, він зауважив, що водяні знаки були досить корисними в програмах, пов'язаних із низьким рівнем безпеки, наприклад, в кабельному телебаченні або вбудовуванні повідомлень, і що нові методи можуть ще покращити роботу алгоритмів водяних знаків. Нарешті, Барні досить цікаво запитав: "Чому ми повинні приховувати інформацію в даних, коли ми могли б легше використовувати заголовки або інші засоби для досягнення тих самих цілей [10]?"

На думку автора, водяні знаки мають важливі недоліки, про що йдеться далі. Хоча водяний знак може бути корисним у певних ситуаціях, його не можна розглядати як засіб для захисту вмісту в довгостроковій перспективі, якщо водяний знак дійсно не захищає всі цінні зміни позначеного об'єкта.

Виявлення дублікатів на основі вмісту

Як вже було сказано, підходи, що базуються на вмісті, залежать від аналізу зображень, а не від вбудови «повідомлень». Більшість існуючих підходів, що базуються на вмісті, базуються на створенні узагальнення зображень, яке називається хеш або дайджест. Хеші згодом використовуються для порівняння зображень із використанням звичайної відстані L_1. Далі ці методи називаються надійним хешуванням. Наскільки відомо автору, небагато методів, заснованих на вмісті, не пов’язані з хешуванням. Однак саме ці роботи представляють особливий інтерес, оскільки ця дипломна робота також спрямована на виявлення дублікатів на основі вмісту, не покладаючись на хешування. Далі методи, не пов’язані з хешуванням, називаються дактилоскопією. Методи відбитків пальців в основному бувають двох видів, як докладно описано в частині 2.3.1, або для опису зображення створюється кілька

хешів, або відстань, яка використовується для порівняння хешів, не базується на звичайній відстані L_1. У першому випадку подібність між зображеннями визначається не відстанню між їх хешами, а скоріше кількістю хешів, які збігаються. У другому випадку метрика відстані часто пристосовується до специфіки пари порівняних зображень. Однак зауважимо, що використання терміна дактилоскопія є характерним для даної тези. Справді, у літературі з обробки сигналів відбитки пальців часто стосуються будь-якої методики виявлення дублікатів на основі вмісту або навіть конкретного застосування водяних знаків, де вбудоване повідомлення використовується для зберігання ідентичності покупця цифрового вмісту.

Загалом підходи до виявлення дублікатів на основі вмісту є більш гнучкими, ніж методи водяних знаків, і не впливають на вміст. Однак вони мають і свої недоліки. Дійсно, хоча методики на основі вмісту можуть бути адаптовані швидше, ніж виявлення дублікатів на основі водяних знаків, наприклад, для протидії новому алгоритму генерації дублікатів, вони більш схильні до зіткнень, або іншими словами, до помилкових виявлень. Наприклад, типові алгоритми водяних знаків досягають коефіцієнтів помилкового виявлення близько одного на мільйон (або більше). З іншого боку, найкращі алгоритми, що базуються на вмісті, досягають лише коефіцієнтів помилкового виявлення близько одного на десятки тисяч.

Вважаю, що методи водяних знаків будуть випереджати показники помилкового виявлення, проте методи, що базуються на вмісті, скоротять розрив. Одна з головних причин полягає у принциповій несхожості двох філософій. Дійсно, при нанесенні водяних знаків вбудоване повідомлення відомо і може генеруватися таким чином, щоб уникнути будь-якої двозначності, навіть у випадках, коли зображення дуже схожі, але різні за змістом. Нічого з цього неможливо за допомогою методів, заснованих на вмісті. Єдина можливість уникнути будь-якої двозначності - це розширені візуальні особливості та сподівання на кращу дискримінаційну(відмінну) силу. З іншого боку, загальноприйнятою практикою є те, що більш насичені функції часто

перетворюються на менш надійні функції, і, отже, можуть змінюватися різко, коли зображення змінюється.

Програми та застосовність

Моніторинг

Моніторинг стосується відстеження зображень, серед іншого, для збору роялті, збору статистичних даних, виявлення порушень авторських прав та незаконного виявлення матеріалів. Ця програма є пасивною, оскільки вона не має прямого впливу на вміст; наприклад, це не заважає показу зображення. Іншими словами, основною функцією цієї програми є спостереження та звітування [11].

Для відстеження використання зображень можуть використовуватися як водяні знаки, так і методи, що базуються на вмісті. Для деяких програм не тільки потрібно виявляти вміст, але й потрібно відстежувати історію розповсюдження. У цьому випадку водяні знаки є єдиним рішенням, оскільки потрібно мати додаткову інформацію. З іншого боку, відстеження застарілого вмісту неможливе за допомогою водяних знаків, і, аналогічним чином, незаконне відстеження зображень не може бути вирішене за допомогою водяних знаків [11]. Справді, водяні знаки вимагають вбудовування перед розповсюдженням, але, в останньому випадку, джерелом керує той, хто має перевагу залишатися анонімним.

Кластеризація

Кластеризація відноситься до перегрупування зображень, які є дублікатами при запиті до бази даних. Як правило, якщо пошукова система зображень, наприклад, зображення Google або Yahoo, запитується за допомогою популярних ключових слів, таких як Ленна чи Брітні Спірс, багато повернених зображень насправді мають однакову або дещо змінену версію. Це зазвичай показано на малюнку 2.1, де зображення Google запитується за ключовим словом Lenna. Отже, було б корисно згрупувати фактичні дублікати під одне зображення, щоб не завалити користувача зайвою інформацією. Для

кластеризації зображень, що повертаються запитом, можна використовувати як водяні знаки, так і виявлення дублікатів на основі вмісту.

Однак використання водяних знаків у цій програмі досить незручне, оскільки для ідентифікації зображення вимагає загальнодоступних водяних знаків. Знову ж таки, використання водяних знаків явно неможливе для застарілого вмісту. З іншого боку, такий тип додатків досить добре підходить до парадигми виявлення дублікатів на основі вмісту.

Пошук версій

Пошук версій - це пошук відповідної версії зображення. Наприклад, припустимо, що у вас є лише мініатюрна версія зображення, яке вам подобається. Отже, було б досить цікаво мати можливість шукати в базі даних зображень, наприклад, зображення Google або Yahoo, запитуючи її за допомогою мініатюри та очікуючи всіх існуючих варіацій цього зображення. Для пошуку версій зображень можна використовувати як водяні знаки, так і виявлення копій на основі вмісту. Знову ж таки, водяні знаки менш підходять для завдання, ніж виявлення дублікатів на основі вмісту, з причин, подібних до тих, що наведені для кластеризації.

2.1.3 Методи, що базуються на вмісті

Виявлення дублікатів на основі вмісту все ще є відносно молодою галуззю обробки сигналів, оскільки перша велика публікація, наскільки відомо автору, датується кінцем дев’яностих. Як наслідок, ще не опубліковано багато робіт, і більшість опублікованих алгоритмів недостатньо зрілі, щоб належним чином оцінити їхню придатність. Дійсно, або продуктивність порівняно низька, або складність методу занадто висока. Крім того, багато звітованих робіт проводять лише побіжне тестування, наприклад, використовуючи лише декілька зображень або обмежену кількість перетворень. Решта частин представляє основні внески у виявлення дублікатів на основі вмісту. Текст

розділений на дві частини, а саме відбитки пальців та надійне хешування, як це визначено в частині 2.2.2.

Ефективність більшості методів виявлення дублікатів на основі вмісту оцінюється з точки зору повноти та точності, що визначаються наступним чином

$$\text{повнота} = \frac{\text{кількість правильно виявлених дублікатів}}{\text{загальна кількість дублікатів}} \tag{2.1}$$

$$\text{точність} \frac{\text{кількість правильно виявлених дублікатів}}{\text{загальна кількість (правильно чи помилково) виявлених дублікатів}} \tag{2.2}$$

Трансформації, що використовуються для створення тестових дублікатів, різняться залежно від твору. У багатьох роботах використовуються еталони водяних знаків, але не всі з них. Зауважте, що стверджувалося, що еталони водяних знаків можуть не адаптуватися для тестування систем виявлення дублікатів на основі вмісту, оскільки вони, як правило, спрямовані на створення дублікатів, вбудовані підписи яких не відповідають фазі оригіналу. Крім того, колекції зображень, що використовуються для оцінки повноти та точності, майже завжди відрізняються, будь то за змістом або за розміром. Нарешті, зверніть увагу, що більшість робіт включають зображення, показані на малюнку 2.2, до набору оригінальних зображень.

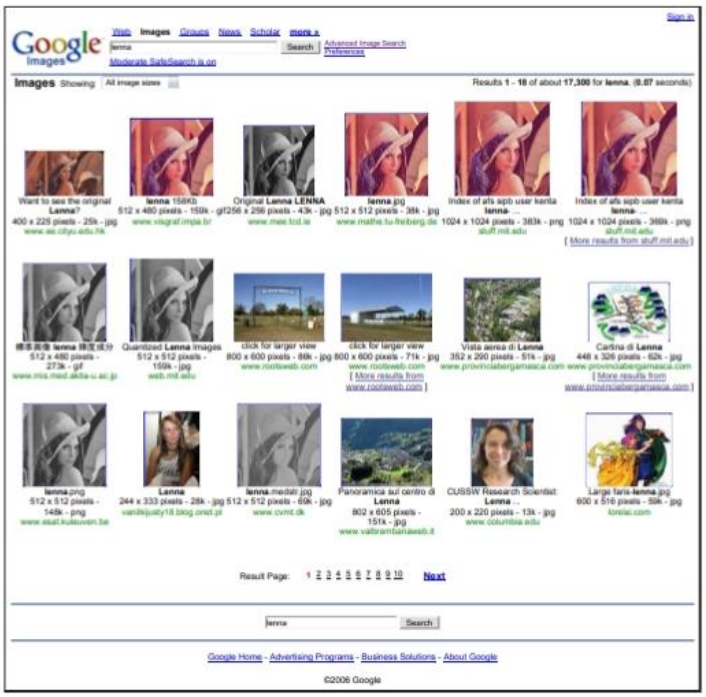


Рисунок 2.1 Зображення Google запитується за ключовим словом „Лена”. На першій сторінці, що повертається цим запитом, десять із шістнадцяти зображень насправді є варіантами оригінального зображення Лени (див. Рисунок 3.2a).



(a) Lenna (b) Pepper (c) Baboon

Рисунок 2.2 Приклад типових оригінальних зображень, що використовуються для тестування методів на основі вмісту.

Метод дактилоскопії

Відбитки пальців стосуються будь-якої техніки, яка використовує узагальнення вмісту зображення, але не покладається на звичайну метрику

відстані для оцінки подібності двох резюме. Межа між дактилоскопією та надійним хешуванням, як було визначено раніше у частині 2.2.2, може бути досить розмитою. Однак для простоти дактилоскопія перегрупує методи, які або генерують кілька хешів для одного зображення, або засновані на нетрадиційних функціях відстані. В обох випадках системи на основі дактилоскопії призводять до більш складних методів індексації, ніж методи, засновані на хешуванні.

Загальна ідея методів дактилоскопії, заснованих на декількох хешах, викладена в даний час, тоді як фактичні методи описані далі [3; 7; 9;]. У цих чотирьох підходах кожен хеш зазвичай описує певну область зображення. Іншими словами, опис зображення робиться багатшим. Кількість регіонів, а також їх локалізація та форми, як правило, залежить від вмісту зображення. Він коливається від декількох хешів до декількох тисяч. Нарешті, два зображення є дублікатами один одного, якщо кількість відповідних хешів перевищує певний поріг. У сімействі підходів до виявлення дублікатів на основі вмісту ці методи на сьогоднішній день є тими, що забезпечують найкращі показники з точки зору точності та повноти. Однак вони часто покладаються на складні особливості і вимагають великої кількості порівнянь. Щоб оцінити, чи два зображення є дублікатами один одного, кожен хеш одного зображення дійсно потрібно порівняти з кожним хешем іншого зображення. З цієї причини методи індексації баз даних відіграють важливу роль у обчислювальній ефективності таких підходів.

Зараз ми звернемо свою увагу на загальну ідею методів відбитків пальців, заснованих на нетрадиційних функціях відстані. Фактичні методи описані далі [11;12]. У цих двох підходах відстань, що використовується для порівняння двох векторів ознак, витягнутих із двох зображень, не є функцією на основі метрики, а є більш складною функцією. Більш конкретно, дублікати зображення не обов'язково лежать в межах гіперсфери, зосередженої на цьому зображенні. Це спостереження призвело до підходів, які можуть мати дуже хороші показники, використовуючи простіші функції щодо методів

дактилоскопії, засновані на декількох представлених раніше хешах. Техніка виявлення дублікатів, запропонована в цій дипломній роботі, насправді базується на нетрадиційній функції відстані.

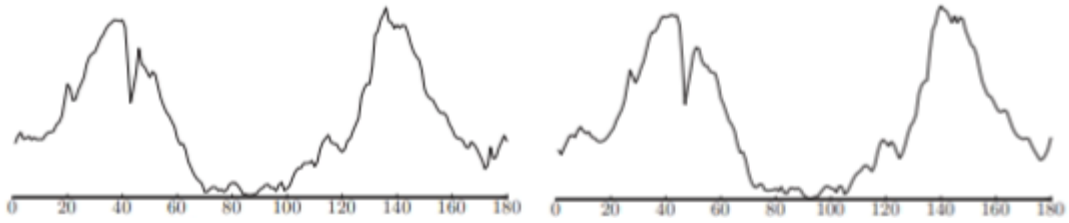
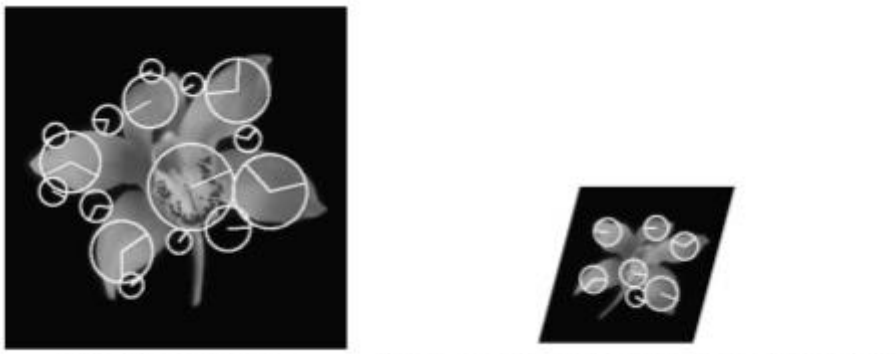


Рисунок 2.3 Приклад хешу для методу, запропонованого в [12]. Ліва фігура відповідає хешу, витягнутому із зображення Лени, тоді як права фігура відповідає фрагменту, витягнутому з поверненої та масштабованої версії. Вісь x дає градуси обертання, тоді як вісь y представляє амплітуду точки середовища.

Відбитки пальців на основі перетворення Радону

Техніка відбитків пальців зображення, розроблена Lefebvre et al.. базується на перетворенні зображення Радона [12;13]. Спочатку алгоритм полягає у обчисленні перетворення Радона зображення [14]. Потім для кожної кутової проекції обчислюється середня точка, інваріантна до перетворення подібності зображення. Хеш нарешті отримується шляхом об'єднання цих незмінних точок. Приклади хешів наведені на рисунку 3.3. Більше того, тип модифікацій, застосованих до зображенні, можна виявити, порівнюючи вихідний хеш із отриманим із модифікованого зображення. Два зображення визначаються як дублікати один одного, спочатку обчислюючи перехресну кореляцію між їхніми хешами, а потім для синхронізації двох хешів використовують положення максимуму. Відстань між двома зображеннями нарешті визначається середньою квадратичною помилкою (MSE) між двома хешами. Щодо класифікації, використаної в цій дипломній роботі, цей підхід

відповідає техніці дактилоскопії, заснованій на нетрадиційній функції відстані, оскільки вимагає обчислення функції перехресної кореляції між зведеннями.



(a) Original image (b) A rotated, scaled, and sheared duplicate

Рисунок 2.4 Приклад ключових точок (КП) у парі повторюваних зображень [11]. КР показані у вигляді білих кіл із вбудованими лініями, що позначають домінуючу орієнтацію, та розміром кола, що позначає шкалу. Багато КП знаходяться в однакових відносних положеннях. Зверніть увагу, що КР, що відповідають меншим масштабам, не представлені (тобто більшість КР відсутні).

Дактилоскопія на основі ключових точок

Ke et al запропонували метод дактилоскопії, заснований на вилученні ознак, які називаються ключовими точками (КР), які є стабільними у масштабному просторі [11]. Зображення, як правило, представлено тисячами КР. Потім тестові зображення класифікуються як дублікати або не дублікати за допомогою локального чутливого хешування, щоб відповідати їх КР до вихідних зображень. Точніше кажучи, безпосередньо не обчислюється жодна відстань, а скоріше кількість відповідних КР, що визначає кількість, якщо два зображення є дублікатами один одного. Що стосується класифікації, використаної в цьому дипломі, цей підхід відповідає техніці дактилоскопії, що базується на множинних хешах, оскільки зображення представлене тисячами локальних зведень.

Ця техніка дактилоскопії в основному заснована на надійності ключових моментів, які є популярними місцевими дескрипторами, представленими в [14]. Детектор КР складається з чотирьох основних етапів, а саме: виявлення максимумів масштабного простору, локалізація КР та призначення орієнтацій. Виявлення максимумів масштабного простору ефективно реалізоване шляхом побудови гауссової піраміди. Пізніше піраміда використовується для виявлення локальних максимумів (званих КР) у послідовності різниці гауссових зображень. На другому етапі локалізація КР уточнюється, а пункти, які виявляються нестабільними, усуваються. На третьому етапі домінуюча орієнтація кожного КР визначається як функція орієнтацій, що знаходяться в навколишній ділянці. Нарешті, останній крок полягає в описі гістограми локальних орієнтацій ділянок, нормованої таким чином, щоб незмінно масштабувати зміни та афінні перетворення. Потім кожне зображення представляється тисячами КР, і тому слід бути особливо обережним при індексації КР. Приклад локалізації КР та домінантних орієнтацій зображений на рисунку 2.4.

Хоча ефективність, отримана цим методом, є найкращою для підходу, що базується на вмісті, на сьогоднішній день, методика спирається на складний дескриптор. Точніше, на реальному комп’ютері потрібно кілька секунд для аналізу зображення, а опис одного зображення складається з тисяч векторів із 150 записів. Це означає, що залежно від вимог системи виявлення дублікатів щодо кількості перевірених зображень в секунду обчислювальна інфраструктура може бути дуже дорогою.

Існують інші роботи, засновані на особливостях, наприклад, [14;15]. На противагу роботі Ке та ін., [10] перетворює набір точок ознак у єдиний двійковий хеш. Крім того, виявлення точок об’єкта набагато простіше, оскільки воно базується на детекторі кутів Харріса [16]. З іншого боку, робота Lejsek та ін. дуже схожий на Ке та ін .. Єдина помітна різниця полягає в тому, що замість дескриптора КР Лоу використовується їх власний дескриптор [9].

Використовуваний дескриптор трохи ефективніший, ніж у Лоу: обчислюється менше KP, при цьому все одно досягаються кращі результати відповідності.

Дактилоскопія на основі функції сприйняття відстані

Техніка дактилоскопії, розроблена Qamra та ін. базується на обчисленні функції перцептивної відстані (DPF) [11;13]. Зауважте, що абревіатура PDF не використовується, щоб уникнути плутанини з функцією щільності ймовірності. Точніше, DPF генерується для кожної пари оригінального та невідомого зображення і вимірює схожість між ними. Загальна ідея підходу полягає в активації різних функцій для різних пар зображень. Отже, при обчисленні відстані враховуються лише найбільш подібні особливості. Що стосується класифікації, використаної в цій дипломній роботі, цей підхід відповідає техніці дактилоскопії, заснованій на нетрадиційній функції відстані, оскільки для обчислення відстані потрібні лише найбільш подібні записи зведення.

Нехай зображення будуть представлені p-мірними векторами ознак, і i-та відстань Δd_i між двома зображеннями буде визначена як абсолютна різниця між i-м елементом. Тоді основна перцептивна функція відстані (DPF) визначається як r-й корінь суми m найменших i-х відстаней $(\Delta d_i)^r$, де r і m - два параметри [13]. Обґрунтування використання DPF ґрунтується на науці когнітивної психології, де показано, що люди роблять висновок про схожість між об'єктами на основі їх подібності, а не з їх несхожості [16]. Кількість m ознак, що використовуються для обчислення відстані, вибирається така, яка в середньому досягає найкращих результатів на тренувальному наборі. Запропонований DPF досягає цікавих результатів, але обмежений тим, що m фіксовано. Дійсно, подібність різних пар об'єктів може залежати від різної кількості ознак. Для подолання цього обмеження Qamra et al. запропонувати три додаткові методи для адаптивного вибору m [17]. Перший метод, який називається пороговим значенням, вибирає всі i-ті відстані нижче фіксованого порогу. Другий метод відбирає DPF відповідно до різних значень m і усереднює їх. Третій спосіб додає ваги кожній ознаці; вага встановлюється як обернена до стандартного

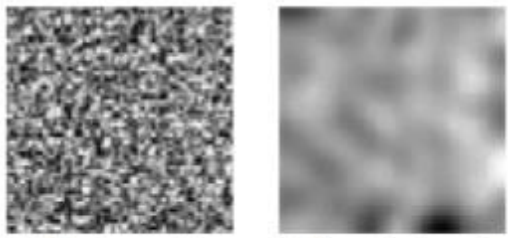
відхилення об’єкта серед подібних зображень. Зверніть увагу, що ці три методи доповнюють один одного і можуть використовуватися разом.

Ефективність методу цікава з точки зору компромісу між точністю та повнотою. Для тестування системи використовується колекція з 20 000 зображень. Серед них 500 зображень випадковим чином відібрані та модифіковані відповідно до 40 дублікатів, використовуються ті самі перетворення, що й у роботі Ке та ін. Ці категорії включають забарвлення, зміни контрасту, обрізання, знімання кольорів, зменшення вибірки (без фільтрування згладжування), перевертання, зменшення глибини кольору, додавання зовнішньої рамки, обертання під прямим кутом, масштабування (з фільтруванням згладжування), зміни насиченості та інтенсивності. Частина 2.3.1 містить докладнішу інформацію про використовувані перетворення. Таким чином, в базі даних індексується 40 000 зображень. Згодом 500 запитаних зображень використовуються для запиту до бази даних, а 40 найбільш подібних зображень підраховуються для визначення кількості помилково спрацьованих результатів, а отже і помилкових негативних результатів. Ефективність нарешті синтезується в кривій точності проти повнотау. Наприклад, повнота 0,9 та 0,8 відповідають точності 0,67 та 0,93 відповідно.

Надійні методи хешування

Хешування відноситься до методів, що використовують єдиний підсумок вмісту зображення, який часто називають дайджестом чи хеш-значенням. При виявленні дублікатів на основі надійного хешування відстань між дайджестами використовується для визначення відповідного співвідношення зображень. Точніше, два зображення є дублікатами один одного, якщо відстань між їх хешами менше певного порогу. Типова відстань базується на L_1-нормі. Концепція хешування дуже схожа на концепцію криптографічних хеш-функцій, яка відображає рядки даних у невелику і постійну кількість бітів. Криптографічні хеш-функції часто і успішно використовуються для автентифікації повідомлень [18]. Однак вони не можуть бути безпосередньо

використані для мультимедійного вмісту, оскільки зображення можуть зазнати досить суворих модифікацій, не змінюючи їх сприймаючих значень. Дійсно, криптографічні функції сконструйовані таким чином, що зміна одного біта повідомлення призводить до зовсім іншого хешу. Цей недолік змусив кілька дослідницьких груп розробити поняття надійного хешування. Також зауважте, що надійне хешування для виявлення дублікатів тісно пов'язане з надійним хешуванням для автентифікації. Дійсно, багато з представлених далі методів хешування мають також компонент безпеки, який спрямований на захист виробленого хешу шляхом рандомізації. Це дозволяє використовувати хеш так само, як і криптографічні хеш-функції. Однак ця особливість насправді не має значення для виявлення дублікатів, як це вивчалось у цьому дипломі.



(a) Random pattern (b) Smoothed version

Рисунок 2.5 Приклад випадкового візерунка та його згладжений варіант [19]. На лівій фігурі зображено випадковий візерунок 64 × 64, а на правій - одна із згладжених версій.

Зусилля щодо стандартизації

Група експертів із рухомих зображень (MPEG) - це робоча група ISO / IEC, яка відповідає за розробку стандартів кодування відео та аудіо. Одним із отриманих стандартів, MPEG-7, є формальна система опису мультимедійного вмісту. В даний час члени MPEG вивчають можливість включення візуального дескриптора в стандартний MPEG-7, який спеціально розроблений для того, щоб служити візуальним ідентифікатором. Іншими словами, вони пропонують

стандартизовану функцію, яка повинна добре працювати для завдання виявлення дублікатів [20;21;31;32;33]. До цього завдання виявлення дублікатів дуже серйозно ставляться в MPEG, оскільки умови тестування та засоби управління зображеннями модифікуються з урахуванням цього нового завдання пошуку.

Запропоновані ознаки базуються на характеристиках та дуже схожі на [20;21;34]. Насправді, цей дескриптор є спрощеною версією, для причин складності, детектора Лоу [22]. Наприклад, локалізація точок об'єкта базується на детекторі кута Гарріса [36], а опис точок об'єкта базується на гістограмі локального градієнта, але для області з фіксованим розміром, а не розміром, який залежить від змісту регіону. Це робота, що все ще триває, і візуальний ідентифікатор, який буде стандартизований, безумовно, покращиться.

У дослідженні MPEG дублікати генеруються відповідно до таких перетворень: зміни яскравості, зміни співвідношення сторін, перетворення кольору на рівень сірого, стиснення JPEG, зменшення глибини кольору, обрізання, вирівнювання гістограми, розмиття, обертання, масштабування, переміщення та перевертання. Потім кожне перетворення параметризується відповідно до трьох рівнів важкості: важких, середніх та легких модифікацій. В даний час запропонований дескриптор надійний проти більшості цих перетворень, за винятком обертання, масштабування та перевертання. У будь-якому випадку він працює краще для завдання виявлення дублікатів, ніж крайовий детектор, який вже присутній у стандарті MPEG-7, але є більш складним.

2.1.4 Підсумок глави

У цій главі представлено сучасний рівень виявлення дублікатів зображень. Спочатку ми розрізняємо дві філософії, а саме водяні знаки та вміст, та описуємо переваги та недоліки кожної з них. В основному, водяні знаки

менш гнучкі, ніж метод, що базується на вмісті, оскільки вимагає модифікації зображення для включення підпису. Точніше, вимога до вбудовування передбачає адаптацію водяних знаків лише в тому випадку, якщо людина має повний контроль над оригінальною ілюстрацією, і означає, що зображення з водяним знаком може бути виявлене лише до тих пір, поки не буде виявлено засіб для видалення підпису. З іншого боку, виявлення дублікатів на основі вмісту є більш гнучким, але ще не таким зрілим, як водяні знаки, з точки зору точності та швидкості повнотау.

Потім ми представили кілька існуючих методів, заснованих на вмісті. Ці методи класифікуються на дві підкатегорії, а саме надійне хешування та дактилоскопія. Міцний підхід хешування полягає в узагальненні зображення за дайджестом, часто двійковим, а потім використовують просту відстань L_1, щоб визначити, чи є два зображення дублікатами один одного чи не пов’язані між собою. З іншого боку, дактилоскопія відноситься до методу, який не може бути класифікований як надійне хешування згідно з попереднім визначенням. Виявляється, більшість методів виявлення дублікатів на основі вмісту мають надійний тип хешування. Однак вони часто покладаються на простіші функції, ніж методи дактилоскопії, і, отже, не так добре працюють. Тим не менше, створений хеш можна легко використовувати для індексації зображень, хоча це не завжди стосується методів дактилоскопії.

Також зазначається, що виявлення дублікатів на основі вмісту все ще залишається недавньою сферою обробки сигналів. Таким чином, поки не існує стандартизованих методів оцінки ефективності методів виявлення дублікатів на основі вмісту. Крім того, існує ще багато різних думок щодо точного визначення проблеми або навіть щодо того, що таке дублікат. Тим не менше, це активна наукова область, яка швидко зростає. Наприклад, це об'єкт пропозиції стандартизації в рамках MPEG-7, принаймні для функцій, які можуть бути спеціально використані для виявлення дублікатів. Дорога ще довга, оскільки багато споріднених проблем потрібно вирішити спочатку, наприклад, визначити стандартний спосіб тестування системи виявлення дублікатів.

Зараз ми синтезуємо існуючі методи виявлення дублікатів на основі вмісту. Синтез можна знайти в таблиці 2.1, де різні методи синтезуються з точки зору пам'яті, складності та продуктивності. Пам'ять відноситься до обсягу пам'яті, необхідного для зберігання опису оригіналу або іншої інформації, необхідної для виявлення його дублікатів. Складність відноситься до обчислювальної складності, необхідної для опису тестового зображення та порівняння опису з описом оригінального зображення. Іншими словами, складність не відноситься до часу, необхідного для навчання системи. Ефективність стосується компромісу між помилково виявленими непов’язаними зображеннями та помилково відхиленими дублікатами. Нарешті, зауважте, що поставлені оцінки - це якісні оцінки автора, отримані завдяки доступній опублікованій інформації.

method	type	memory	complexity	performance
[Lefebvre *et al.*, 2003]	fingerprinting	+	+	−
[Ke *et al.*, 2004]	fingerprinting	−	−	+ + +
[Hsu and Lu, 2004]	fingerprinting	+	+	++
[Qamra *et al.*, 2005]	fingerprinting	+	++	++
[Wang *et al.*, 2006]	fingerprinting	+	++	−
[Fridrich, 2000]	hashing	+ + +	+	+
[Venkatesan *et al.*, 2000]	hashing	++	+	+
[Kim, 2003]	hashing	++	+ + +	+
[Seo *et al.*, 2004]	hashing	+ + +	+	+

Рисунок 2.7 Синтез сучасних методів виявлення дублікатів. Ця таблиця узагальнює деякі сучасні методи. Позначки +, ++, + + + відносяться, відповідно, так собі, добре і відмінно, тоді як - позначає недолік. Порядок появи методів такий самий, як у частині 3.3.

2.2 Фреймворк для виявлення копій зображень на основі вмісту

У першій частині цього розділу ми визначаємо загальну основу для систем виявлення дублікатів на основі вмісту. Запропонована структура починається з простої математичної моделі дублікатів зображення. Модель

визначає підпростір, що охоплюється дублікатами оригіналу, і представлена в частині 2.2.1. Ця модель дозволяє отримати певне розуміння того, як організовані дублікати, і дає корисні вказівки щодо того, як створити ефективну систему виявлення дублікатів. Другим елементом фреймворку є загальна система виявлення дублікатів оригінальних зображень; це враховано в частині 2.2.2. Проаналізовано два випадки. Перший випадок стосується системи, яка виявляє дублікати одного зображення, тоді як другий випадок стосується більш загальної системи, яка одночасно виявляє дублікати кількох зображень. Наступним елементом основи є методологія оцінки, про яку повідомляється в частині 2.2.3, де ми детально описуємо тестові зображення та показники помилок, що використовуються для оцінки ефективності запропонованої системи виявлення дублікатів.

У другій частині ми даємо огляд запропонованої системи виявлення дублікатів у частині 2.2.4. Крім того, у цій частині також представлені загальні компоненти запропонованої системи виявлення дублікатів. Точніше, вона включає операції попередньої обробки, застосовані до зображення, та подальшу процедуру вилучення візуальних функцій низького рівня.

2.2.1 Модель дублікатів зображення

Загальна ідея підходу, запропонованого в цій дипломній роботі, полягає в оцінці області простору зображення, в якій лежать дублікати конкретного зображення. Потім дублікати можна легко виявити, стверджуючи, чи знаходиться тестове зображення всередині або за межами цієї конкретної області.

Відносно важко уявити криву, вбудовану в простір зображення, оскільки цей простір має велику кількість вимірів. З іншого боку, для опису зображення можна використовувати відносно обмежену кількість візуальних особливостей. Ця ідея проілюстрована на малюнку 2.2.1 щодо двох візуальних особливостей

та операції зміни розміру. Далі модель дублікатів зображення дається для простору зображень, але її можна легко розширити до простору об’єктів. Далі ми формалізуємо цю ідею і поширюємо її на дублікати дублікатів.

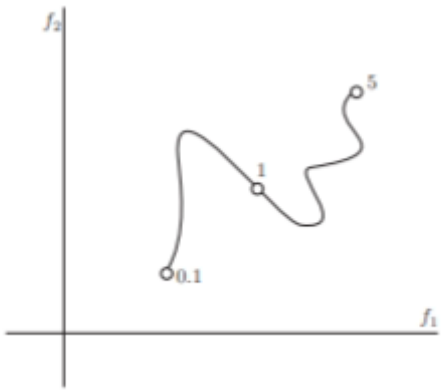


Рисунок 2.8 Крива, визначена зображенням та його дублікатами. На малюнку показано двовимірний простір об’єктів, який ілюструє дублікати, отримані в результаті модифікації вихідного зображення плавним перетворенням, наприклад, зменшення розміру в коефіцієнт від 0,1 до 5, у цьому випадку коефіцієнт зміни розміру одиниці відповідає вихідному зображення.

Припустимо, що зображення є гладкими двовимірними функціями і, отже, що простір зображення є простором гладких функцій. Далі ми припускаємо, що розглянуті перетворення зображень є гладкими функціоналами, визначеними на цьому просторі. Тоді набір, що містить дублікати вихідного зображення I, можна визначити наступним чином

$$D(I) = \{f_i(I, \rho): \rho \in C_i, i = 1,2, \dots, N\} \quad (2.3)$$

де f_i (I, p) - N, що вважаються функціоналами перетворення, p означає параметр кожного функціоналу, а набір C_i дає відповідні параметризації або, іншими словами, можливі значення параметрів. Для кращого інтуїтивного відчуття давайте розглянемо, що f_1 (I, p) відповідає функціоналу або операції

перетворення розміру. У цьому випадку p означає коефіцієнт масштабування, а C_1 відповідає діапазону дозволених коефіцієнтів масштабування; наприклад C_1 задається інтервалом [0.1, 5] для прикладу, представленого раніше. Вихідне зображення I неявно є частиною D (I), оскільки ми припускаємо, що для будь-якого перетворення f (·, ·) існує інваріантна параметризація p така, що I = f (I, p). Наприклад, у випадку операції зміни розміру це означає, що відповідна параметризація C_1 містить дійсне число один, що створює повторюване зображення, рівне вихідному зображенню.

Крім того, дублікати дублікатів можна по черзі вважати дублікатами оригіналу, в цьому випадку також включаються інші криві, що проходять через кожен дублікат. Дублікат набору E_n (I) для композицій до n-рівня може бути рекурсивно визначений

$$\varepsilon_n(I) = \{ D(J) : J \in \varepsilon_{n-1}(I) \} \tag{2.4}$$

$$\varepsilon_1(I) = D(I) \tag{2.5}$$

Зауважимо, що $\varepsilon_n(I) \supseteq \varepsilon_n - 1$ (I) $\supseteq \cdots \supseteq \varepsilon_1(I)$через існування інваріантної параметризації.

Сукупність $\varepsilon_n(I)$ є складним об'єктом для сприйняття. Для аналізу $\varepsilon_n(I)$ ми спочатку вводимо спрощення. Справді, давайте тепер розглянемо, що дублікати, що виникають на основі n-рівня складу, можуть бути виражені одним функціональним g_n (I, p). Перша змінна I є вихідним зображенням, а інша змінна p є вектором параметрів, який контролює дублікат аспекту, наприклад p_1 може бути коефіцієнтом масштабування, а p_2 кутом повороту. Такий функціонал може бути рекурсивно побудований за допомогою введених раніше функціоналів перетворення f_i (I, p). Точніше

$$g_n(I, \rho) = \mathfrak{f}_n(g_{n-1}(I, \rho), \rho_n) \tag{2.6}$$

$$g_n(I, \rho) = \mathfrak{f}_1(I, 1) \tag{2.7}$$

Зауважимо, що порядок операцій можна змінити, переставляючи індекси і-перетворень f_i (I, p). У цьому спрощеному випадку набір дублікатів для n-рівня композиції задано як

$$Ƒ(I, n) = \{g_n(I, \rho): \rho \in C_1 \times C_2 \times \ldots \times C_n\} \quad (2.8)$$

Таким чином, дублікат набору F (I, n) визначається обмеженою гладкою високовимірною поверхнею або гладким колектором, вбудованим у простір зображення. Внутрішня розмірність колектора обмежена зверху n, кількістю розглянутих композицій, оскільки колектор створюється функцією, керованою n + 1 параметрами, і одним з них є вихідне зображення.

Крім того, можна пов'язати колектор Ƒ(I, n) із більш складним об'єктом E_n(I), визначеним вище. Дійсно, давайте тепер розглянемо, що кількість композицій n дорівнює кількості перетворень N. Оскільки різні множини Ƒ(I, n) можуть бути побудовані для різних порядків операцій, ми отримуємо

$$\varepsilon_N(I) = \cup_\nu Ƒ(I, \nu) \quad (2.9)$$

де ν - перестановка перших N натуральних чисел, $U_ν$ означає об’єднання на всіх можливих перестановках, а F(I, ν) позначає набір повторень, як визначено у рівнянні (2.2.6), але з порядком операцій, модифікованим відповідно до перестановки ν. Цей результат означає, що E_N (I) задається об'єднанням N! гладких різноманітностей.

Тепер ми аналізуємо вплив зміни порядку операцій на дублікат набору $\varepsilon_N(I)$. Для цього ми припускаємо, що зміна порядку операцій змінює отриманий дублікат, але не різко. Нехай d (·, ·) - функція відстані в просторі зображення і визначимо

$$\varepsilon(I) = \max_\rho \max_{\nu_1 \nu_2} d(g_N(I, \rho, \nu_1), g_N(I, \rho, \nu_2)) \quad (2.10)$$

де $ν_{1,2}$ - перестановки перших N натуральних чисел, а g_N (I, p, ν) означає функціонал єдиного перетворення, визначений вище, але з порядком операцій, модифікованим відповідно до перестановки ν. Тоді максимізація на $ν_1$ і $ν_2$ дає

максимально можливу відстань між дублікатами для даної параметризації p. Нарешті, значення ξ (I) дає найбільшу таку відстань серед усіх можливих параметрів. Тепер визначимо великомірний об’єм V(I, ν) на основі різноманіття, породженого довільним порядком операцій ν.

$$V(I,\nu) = \{J : d(J,K) \leq \xi(I), K \in \mathcal{F}(I,\nu)\} \quad (2.11)$$

Для будь-якого порядку операцій, визначеного перестановкою μ, ми маємо V (I, ν) ⊃ F (I, μ). Цей результат означає, що дублікати зображення можуть бути укладені у великомірний об'єм, який має товщину 2ξ(I). Товщина пропорційна впливу порядку операцій. Наприклад, якщо зміна порядку операцій не змінює результуючого дубліката, тоді дублікати зображення лежать на гладкому колекторі, внутрішнє число розмірностей якого обмежене верхньою частиною N, кількість розглянутих перетворень.

У реальному світі багато припущень не виконуються. Наприклад, зображення - це не плавні сигнали, а скоріше дискретизовані та просторово обмежені сигнали. Крім того, перетворення можуть бути непростими; Типовими прикладами негладких перетворень є стискання або обрізання експертної групи спільних зображень (JPEG). Тим не менше, модель, розроблена в цій частині, залишається корисною, оскільки дає підказки щодо того, як розробити ефективну систему виявлення дублікатів. Посилаючись на цю модель далі, ми часто називаємо її підпростором, охопленим дублікатами оригіналу.

2.2.2 Загальна система виявлення дублікатів

У цій частині пропонується загальна система виявлення дублікатів. Загальна система складається з системи, яка одночасно виявляє дублікати кількох вихідних зображень. Однак спрощена версія загальної системи, де розглядається лише одне оригінальне зображення, вперше представлена в

частині 2.2.2.1. Потім система виявлення кількох оригінальних дублікатів представлена в частині 2.2.2.2. Не тільки система виявлення дублікатів, запропонована в цьому дипломі, базується на загальній системі, але також спрощена версія є одним із ключових компонентів запропонованої системи. Нарешті, спрощені та загальні системи служать основою для розробки адекватних методів оцінки їх ефективності, як це зроблено у частині 2.3.

Загальне виявлення дублікатів - система оригінального зображення

Виявлення дублікатів одного оригіналу можна змоделювати наступним чином. Ми розглядаємо систему, налаштовану на виявлення дублікатів конкретного вихідного зображення O. Цю систему виявлення дублікатів можна розглядати як двійковий класифікатор, який відображає тестове зображення T в один із двох класів. Точніше, мітка +1 відповідає класу «тестове зображення T є дублікатом оригінального зображення O», тоді як мітка -1 означає клас «T не має відношення до O» Таку систему можна звести до двійкової функції $d_O^1(\cdot, u) \in \{-1, +1\}$, де u - параметр, що контролює селективність системи. Точніше, $d_O^1(T, u)$ дорівнює +1, якщо тестове зображення T оцінюється як дублікат O, і -1, якщо T і O вважаються не пов’язаними. Виявлення дублікатів одного оригіналу показано на малюнку 2.2.2.

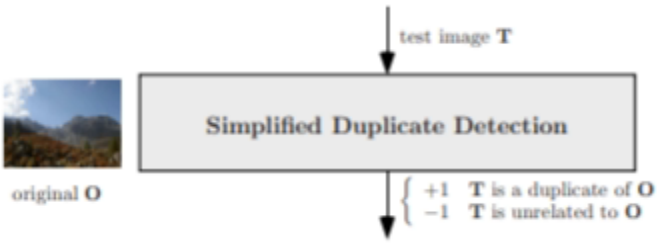


Рисунок 2.9 Спрощена система виявлення дублікатів (один оригінал).

Механіку системи можна визначити наступним чином. Основна ідея полягає в тому, щоб оцінити ймовірність Pr {T ~ O}, що тестове зображення T є дублікатом вихідного зображення O. Потім можна отримати рішення, порівнявши оцінену ймовірність Pr {T ~ O} з фіксованим порогом $u \in [0, 1]$.

Якщо ймовірність перевищує поріг, тоді тестове зображення вважається дублікатом вихідного зображення; інакше обидва зображення розглядаються як не пов'язані між собою. Нарешті, $d_O^1(\cdot, u)$ формально задано формулою

$$d_O^1(T, u) = 2 \cdot \left(I_{\{x:x>u\}}(\Pr\{T \sim O\}) - \frac{1}{2} \right) \quad (2.12)$$

де $u \in [0, 1]$ - поріг, O - оригінальне зображення, T - тестове зображення, $\Pr\{T \sim O\}$ - передбачувана ймовірність того, що T - дублікат O, а $I_A(x)$ - показник функції. Нагадаємо, що $I_A(x)$ дорівнює одиниці, якщо $x \in A$, і нулю в іншому випадку.

Загальне виявлення дублікатів - система кількох оригінальних зображень

Виявлення дублікатів кількох оригінальних зображень можна змоделювати наступним чином. Тепер ми розглянемо систему, налаштовану на одночасне виявлення дублікатів будь-якого оригіналу серед набору конкретних оригінальних зображень O. Кожному елементу $\cdot$ O, позначеному далі $O(\cdot)$, відповідає конкретне оригінальне зображення. Цю систему виявлення дублікатів можна розглядати як багатокласний класифікатор, який відображає тестове зображення T в один із класів N + 1, де N дорівнює $|O|$, кількості вихідних зображень. Точніше, N класів, позначених +1, +2,. . . , + $|O|$, відповідають випадку «тестове зображення T є дублікатом відповідних вихідних зображень». З іншого боку, залишений клас, позначений −1, означає випадок "T не пов'язаний з жодним зображенням серед O." Така система зводиться до цілочисельної функції $d_O^N(\cdot, u) \in \{-1, +1,. . ., |O|\}$ де u - параметр, що контролює вибірковість системи. Точніше, $d_O^N(T, u)$ дорівнює цілому додатньому числу i, якщо тестове зображення T оцінюється як дублікат вихідного зображення O (i) та −1, якщо T вважається не пов'язаним з жодним із зображень, що містяться в O. Виявлення дублікатів кількох оригіналів показано на рисунку 4.3.

Механіку системи можна визначити наступним чином. Основна ідея полягає в оцінці набору ймовірностей {Pr {T ~ O (i)}} $^{N}_{i}$=1, де кожен елемент Pr {T ~ O (i)} - це оцінка ймовірності того, що T є дублікатом відповідного оригінального зображення O (i). Іншими словами, ймовірності можна оцінити за допомогою окремих оригінальних повторюваних детекторів, представлених у попередньому частині. Потім можна отримати рішення, порівнявши найбільшу ймовірність, що міститься у згаданому наборі, з фіксованим порогом u. Якщо ймовірність перевищує поріг, тоді тестове зображення T вважається дублікатом відповідного оригінального зображення, тоді як інакше T розглядається як не пов'язане з будь-яким з оригінальних зображень. Нарешті, d^{N}_{O} (·, u) формально задано формулою

$$m = \arg \max_{i=1,\dots,N} \rho_{T \sim O_{(i)}} \tag{2.12}$$

$$d^{N}_{O}(T,u) = m \cdot I_{\{x:x>u\}}(\Pr\{T \sim O(m)\}) - I_{\{x:x \le u\}}(\Pr\{T \sim O(m)\}) \tag{2.13}$$

де u ∈ [0, 1] - поріг, O - набір оригінальних зображень, T - тестове зображення, Pr {T ~ O (i)} - передбачувана ймовірність того, що T є дублікатом відповідного вихідного зображення O (i), а I_A (x) - це функція індикатора. Якщо найбільша оцінна ймовірність менша за порогову, перший доданок у правій частині рівняння (4.12) тоді дорівнює нулю, а другий доданок дорівнює мінус одиниці. З іншого боку, якщо найбільша оцінна ймовірність більша за порогове значення, тоді другий доданок у правій частині рівняння (4.12) дорівнює нулю, а перший доданок дорівнює мітці передбачуваного оригіналу тесту зображення.

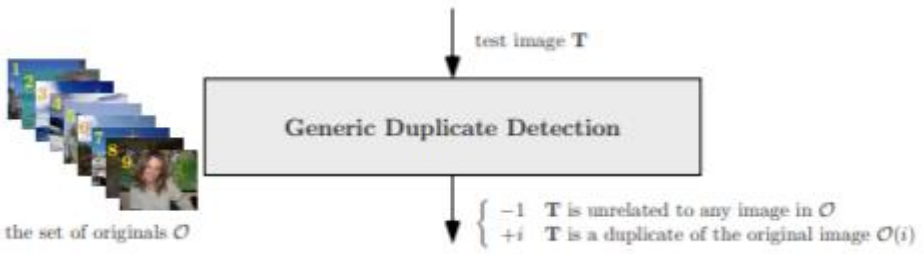


Рисунок 2.10 Загальна система виявлення дублікатів (кілька оригінальних зображень).

2.2.3 Методи оцінки результативності

У цій частині ми представляємо зображення та метрики, що використовуються для оцінки ефективності алгоритмів виявлення дублікатів, запропонованих у наступних главах. Тестові зображення представлені в частині 4.3.1. Потім дві метрики визначені в частині 4.3.2. Один показник використовується для оцінки продуктивності системи виявлення дублікатів оригінального зображення, тоді як інший використовується для системи виявлення копій кількох оригінальних зображень.

Тестові зображення

Для оцінки ефективності запропонованих систем використовуються ті самі колекції зображень, що й у [37]. Перша колекція містить 18 785 фотографій, включаючи, але не обмежуючись ними, пейзажі, тварин, споруди та людей. Розміри зображення та співвідношення сторін змінюються, наприклад, 900 × 600, 678 × 435 або 640 × 480 пікселів. В основному це кольорові зображення, за винятком близько тисячі зображень із рівнями сірого. Друга колекція містить фотографії 9000 картин. Використання колекцій з різноманітним вмістом дозволяє оцінити ефективність алгоритмів виявлення дублікатів у різних ситуаціях. Наприклад, перша колекція містить фотографії, що охоплюють широкий спектр сцен, тоді як друга колекція містить дуже схожі зображення за кольорами та фактурами.

Колекції випадковим чином поділяються на два взаємовиключні підмножини O та F. Набір O представляє оригінали та містить 200 зображень, а набір F - зображення, які використовуються для оцінки частоти помилок помилково спрацьовуючих результатів системи.

categories	♯, parameterisations
Colourising	3, Tint the red, green, or blue channel by 10%
Contrast changes	2, Increase or decrease the contrast[a]
Cropping	4, Crop by 5, 10, 20 and 30%
Despeckling	1, Apply ImageMagick's despeckling operation
Downsampling	6, Downsample by 10, 20, 30, 40, 50, 70 and 90%[b]
Flipping	1, Flip along the horizontal axis
Colour depth reduction	1, Reduce the colour palette to 256 colours
Outer frame	4, Add an outer frame of 10% the image size
Rotation	3, Rotate by 90, 180 and 270°
Scaling	6, Scale up by 2, 4, 8 times, and down by 2, 4, 8 times[c]
Saturation changes	6, Change the values of the saturation channel by 70, 80, 80, 90, 110, 120 and 130%
Intensity changes	4, Change the intensity channel by 80, 90, 110 and 120%

[a] using ImageMagick's [Still, 2005] default parameter
[b] without antialiasing filtering
[c] with antialiasing filtering

Рисунок 2.11 Тест набору повторюваних зображень Qamra. Цей набір тестів містить дублікати, запропоновані в [22] та використані також у [38]. Він імітує трансформації, які часто зустрічаються при публікації зображень в Інтернеті.

Повний набір тестів складається з двох компонентів: набору зображень та набору міток. Набір зображень, що позначається T, дається об'єднанням неспоріднених зображень F і будь-якого з дублікатів набору Qamra або StirMark. Набір міток, позначений L, пов'язує кожне зображення в T з міткою; а саме 0 для непов'язаних зображень та різні додатні числа (+1, +2, ..., + | O |) для дублікатів кожного оригінального зображення.



(a) Original

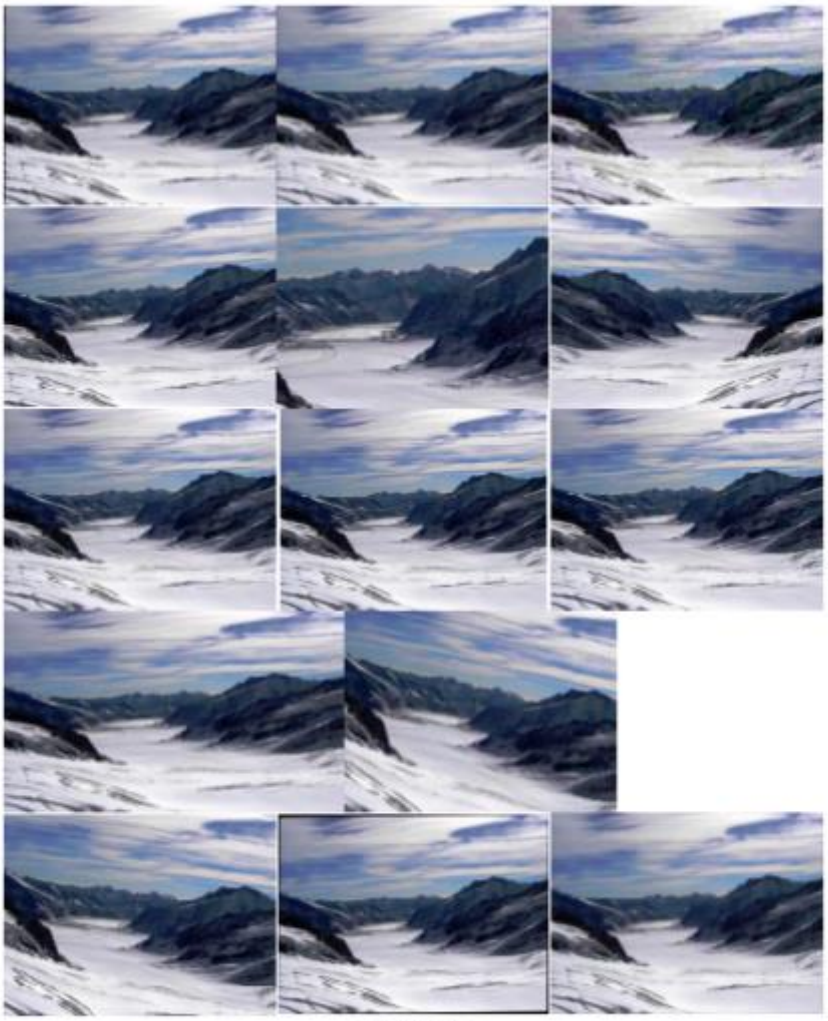


(b) Duplicates

Рисунок 2.12 Приклади тестових дублікатів, сформованих тестом Qamra. У кожній категорії є один приклад, що повторюється, порядок (зліва направо, зверху вниз) такий самий, як у таблиці 4.1. Кожне зображення має розмір, щоб мати однакову висоту. Для інформації фотографія зроблена на вершинах Нендазу - Швейцарія.



(a) Original



(b) Duplicates

Рисунок 2.13 Приклади тестових дублікатів, сформованих контрольним тестом StirMark. У кожній категорії є один приклад, що повторюється, порядок (зліва-направо, зверху вниз) такий самий, як у таблиці 2.2.2. Кожне зображення має розмір, щоб мати однакову висоту. Для інформації фотографія була зроблена з Jungfraujoch - Швейцарія.

categories	♯, parameterisations
Median filtering	3, filter of size 2×2, 3×3 and 4×4
Gaussian filtering	1, approximate filter of size 3×3
JPEG compression	12, JPEG compression with quality factors 90, 80, 70, 60, 50, 40, 35, 30, 25, 20, 15 and 10
Shearing	6, shearing in (X, Y) directions by (0, 1), (0, 5) (1, 0), (5, 0), (1, 1) and (5%, 5%)
Cropping	9, centred cropping by 1, 2, 5, 10, 15, 20, 25, 50 and 75%
Flipping	1, horizontal flip
Scaling	6, scaling by factors 0.5, 0.75, 0.9, 1.1, 1.5 and 2
Line removal	5, removal of (n columns, m rows): (1, 1), (1, 5), (5, 1), (5, 17) and (17, 5)
Random bending	1, 'StirMark' random geometric distortions
Aspect ratio	8, change aspect ratio of X(Y) by a factor 0.8, 0.9, 1.1 and 1.2
Rotation	16, rotations by -2, -1, -0.75, -0.5, -0.25, 0.25, 0.5, 0.75, 1, 2, 5, 10, 15, 30, 45 and 90°
Rotation/scaling	16, same as above but followed by scaling
Linear transform	3, general linear geometric transformation $\mathbf{c}' = \mathbf{T}\mathbf{c}$ [a]
FMLR	3, frequency mode Laplacian removal attack

[a] values of $\mathbf{T}$: $\begin{pmatrix} 1.010 & 0.013 \\ 0.009 & 1.011 \end{pmatrix}$, $\begin{pmatrix} 1.007 & 0.010 \\ 0.010 & 1.012 \end{pmatrix}$, and $\begin{pmatrix} 1.013 & 0.008 \\ 0.011 & 1.008 \end{pmatrix}$

Рисунок 2.14 Тестовий набір повторюваних зображень StirMark. Цей набір тестів містить дублікати, запропоновані в [39] та використані при оцінці більшості алгоритмів водяних знаків. Він імітує трансформації, які часто зустрічаються при копіюванні зображень.

Показники ефективності

Існують різні способи оцінки роботи системи виявлення дублікатів. Наприклад, можна виміряти компроміс між помилковими спрацьовуваннями та помилковими негативними показниками, або між точністю та повнотаанням. Для отримання додаткової інформації щодо точності та повнотаання читача див. Частина 3. Далі ми використовуємо парадигму хибнопозитивних та хибнонегативних показників, оскільки проблема виявлення дублікатів розглядається в цьому дипломі з точки зору класифікації та не з точки зору пошуку.

Як вже згадувалося раніше, продуктивність системи виявлення дублікатів може бути оцінена шляхом компромісу між коефіцієнтом помилок помилкових спрацьовувань та помилкових негативних результатів. Помилкова позитивна помилка - це тестове зображення, яке, за оцінками, є копією оригіналу, але не є.

І навпаки, помилково негативна помилка - це тестове зображення, яке є дублікатом оригіналу, але не виявляється як таке. Далі істинна мітка класу загального тестового зображення позначається c_t, а оціночна мітка класу - c_e. Хоча помилково позитивні помилки можуть виникати щоразу, коли тестове зображення оцінюється як дублікат, $c_e > 0$, помилково негативні помилки трапляються лише тоді, коли тестове зображення дійсно є дублікатом, $c_t > 0$. В обох випадках помилки означають, що передбачувана мітка та справжній ярлик відрізняється. Точніше, за умови справжнього класу c_t та оціночного класу c_e трапляється помилка, якщо функція e (c_t, c_e), надана далі, дорівнює одиниці.

$$e(c_t,c_e) = 1 - I_{\{c_t\}}(c_e) \tag{2.14}$$

де I_A (x) - функція індикатора. Як було сказано раніше, помилково позитивна помилка трапляється, якщо помилка є і якщо оцінювана мітка більше нуля. Відповідно, для істинного класу c_t та оціночного класу c_e, помилково позитивна помилка трапляється, якщо

$$fp(c_t,c_e) = e(c_t,c_e) \cdot I_N \cdot (c_e) \tag{2.15}$$

дорівнює одиниці. Подібним чином помилково негативна помилка означає, що

$$fn(c_t,c_e) = e(c_t,c_e) \cdot I_N \cdot (c_t) \tag{2.16}$$

дорівнює одиниці. Нарешті, жодна помилка не означає, що не існує помилково позитивних або помилково негативних помилок, або, іншими словами, як fp (c_t, c_e), так і fn (c_t, c_e) дорівнюють нулю.

На відміну від кривих ROC, криві DET представляють частоту помилок фальшивих негативів як функцію коефіцієнта помилок фальшивих негативів. Оскільки обидві осі відповідають вимірюванню похибок, вони обидві можуть використовувати логарифмічну шкалу. Інтерпретація кривих DET аналогічна інтерпретації кривих ROC: класифікатор X є більш точним, ніж класифікатор Y, коли його крива DET нижче, ніж Y. Точна побудова кривої DET залежить від

використовуваної системи виявлення дублікатів і є подані у наступних двох підчастинаах.

Показники ефективності - система оригінального зображення

Для однієї оригінальної моделі виявлення дублікатів частота помилок помилкових спрацьовувань та помилкових негативних результатів дорівнює $p_{FP} = \Pr\{c_e \neq c_t \mid c_t < 0\}$ та $p_{FN} = \Pr\{c_e \neq c_t \mid c_t > 0\}$ відповідно. Вони розвиваються наступним чином

$$\rho_{FP} = \Pr\{c_e = +1 | c_t = -1\} \quad (2.17)$$

$$\rho_{FN} = \Pr\{c_e = -1 | c_t = +1\} \quad (2.18)$$

оскільки в двійкових детекторах відомі лише два класи.

Отже, оцінка частоти помилок помилково спрацьовуючих даних для порогового значення селективності u та даного вихідного зображення O (n), дається формулою

$$\hat{\rho}_{FP}(O(n), u) = \frac{1}{\sum_{j=1}^{|\mathcal{L}|} I_{\{-1\}}(\mathcal{L}(j))} \sum_{\substack{i=1 \\ \text{за умови } \mathcal{L}(i) = -1}}^{|T|} fp\left(-1, \mathrm{d}^{\mathrm{N}}_{\mathrm{O}\,(n)}(T(i), u)\right) \quad (2.19)$$

де T - набір тестових зображень, визначених у частині 4.3.1, L - відповідний набір міток, $\mathrm{d}^{1}_{\mathrm{O(n)}}$ (T (i), u) - функція (визначена в частині 4.2.1), яка оцінює мітку класу тестового зображення T (i) щодо вихідного зображення O (n), fp (·, ·) - це функція, яка вказує на помилково позитивну помилку та визначена в попередньому частині. Аналогічним чином, оцінки коефіцієнта помилок помилкових негативів, для порогового значення селективності u та даного вихідного зображення O (n), даються формулою

$$\hat{\rho}_{FN}(O(n), u) = \frac{1}{\sum_{j=1}^{|\mathcal{L}|} I_{\{-1\}}(\mathcal{L}(j))} \sum_{\substack{i=1 \\ \text{за умови } \mathcal{L}(i) = -1}}^{|T|} fn\left(+1, \mathrm{d}^{\mathrm{N}}_{\mathrm{O}\,(n)}(T(i), u)\right) \quad (2.20)$$

Для оцінки продуктивності системи, яка виявляє дублікати одного оригіналу, ми використовуємо кілька детекторів. Кожен детектор

налаштований на певне зображення O (n). Компромісний алгоритм між помилковими позитивними та помилковими негативними показниками зведений до єдиної кривої DET, побудованої наступним чином. Для кожного вихідного зображення O (n) формується крива DET шляхом збору оціночних ймовірностей $\hat{p}_{FP}$ (n, u) та $\hat{p}_{FN}$ (n, u) для різних значень порога u.

Усі криві нарешті синтезуються в єдину криву DET, позначену $\overline{DET}$, за допомогою вертикального усереднення, алгоритм 5 у [40]. У процедурі вертикального усереднення коефіцієнт неправильних негативних помилок отримується шляхом усереднення коефіцієнтів неправильних негативних помилок даних, наданих різними кривими DET, при однаковій частоті неправильних позитивних помилок даних. Це означає, що робоча точка на кривій $\overline{DET}$ відповідає порогам, які, можливо, різні для кожного детектора. На практиці для пошуку правильних порогових значень у залежності від обраної робочої точки можна використовувати таблицю пошуку.

Використання вертикального усереднення на кривих DET дозволяє отримати оцінки оптимальних показників ансамблю двійкових детекторів, незалежно від методу їх комбінування.

Показники ефективності - система кількох оригінальних зображень

Для моделі виявлення дублікатів кількох оригіналів частота помилок помилкових спрацьовувань та помилкових негативних результатів дорівнює $p_{FP} = \Pr\{c_e \neq c_t, c_e > 0\}$ та $p_{FN} = \Pr\{c_e \neq c_t \mid c_t > 0\}$ відповідно. Використовуючи попередні ймовірності $p_{\cdot} = \Pr\{c_t = \cdot\}$, коефіцієнт помилок помилкових позитивних спрацьовувань стає

$$\rho_{FP} = \sum_{i=-1}^{|O|} \Pr\{c_e \neq c_t,\ c_e > 0 | c_t = i\}_{\rho_i} \quad (2.21)$$

$$\rho_{FP} = \sum_{i=-1}^{|O|} \Pr\{c_e \neq c_t,\ c_e > 0 | c_t = i\}_{\rho_i} + \Pr\{c_e > 0 | c_t = -1\}_{p_{-1}} \quad (2.22)$$

І навпаки, помилково негативних розвивається наступним чином

$$p_{FP} = \frac{1}{\sum_{i=1}^{|O|} p_i} \sum_{i=-1}^{|O|} \Pr\{c_e \neq c_t | c_t = i\}_{p_i} \quad (2.23)$$

$$p_{FP} = \frac{1}{1-p_{-1}} \sum_{i=1}^{|O|} \Pr\{c_e \neq c_t | c_t = i\}_{p_i} \quad (2.24)$$

$$p_{FP} = \frac{1}{1-p_{-1}} \sum_{i=-1}^{|O|} \Pr\{c_e \neq c_t,\ c_e > 0 | c_t = i\} + \Pr\{c_e = -1 | c_t = i\}_{p_i} \quad (2.25)$$

де рівняння (4.23) виходить з того факту, що події $c_t = +1$ до $c_t = |O|$ утворюють частина події $c_t \neq -1$, аналогічно подіям $c_e = -1$ і $c_e \neq -1$ утворюють розбиття простору результатів. Використовуючи теорему Байєса, це приводить до рівняння (4.24). Визначимо такі величини

$$p_{D \leftrightarrow D} \equiv \frac{1}{1-p_{-1}} \sum_{i=1}^{|O|} \Pr\{c_e \neq c_t, c_e > 0 | c_t = i\}_{p_i} \quad (2.26)$$

$$p_{D \leftrightarrow U} = \frac{1}{1-p_{-1}} \sum_{i=1}^{|O|} \Pr\{c_e = -1 | c_t = i\}_{p_i} \quad (2.27)$$

$$p_{U \leftrightarrow D} = \Pr\{c_e = -1 | c_t = -1\} \quad (2.28)$$

Їх можна інтерпретувати наступним чином $p_{D \leftrightarrow D}$ дає ймовірність того, що дублікату присвоєно неправильний оригінал. Подібним чином $p_{D \leftrightarrow U}$ дає ймовірність того, що фактичний дублікат оцінюється як не пов'язаний з будь-якими оригінальними зображеннями. Нарешті, $p_{U \to D}$ дає ймовірність того, що непов'язане зображення оцінюється як дублікат якогось оригіналу. Тоді коефіцієнт помилок помилкових позитивних спрацьовувань та помилкових негативних результатів можна виразити через $p_{D \leftrightarrow D}$, $p_{D \to 0}$ та $p_{U \to D}$

$$p_{FP} = p_{D \leftrightarrow D} (1 - p_{-1}) + p_{U \to D} p_{-1} \quad (2.29)$$

$$p_{FN} = p_{D \leftrightarrow D} + p_{D \to U} \quad (2.30)$$

Тепер ми можемо дати оцінку ймовірності помилок. Перше зауважимо, що пріори, як правило, невідомі. З іншого боку, можна припустити, що тестове зображення частіше є не пов'язаним із ним зображенням, ніж дублікатом

оригіналу; таким чином $p_{-1} \gg \sum_{i=1}^{|O|} p_i$. Це означає, що $p_{-1} \approx 1$, і навпаки, що $1 - p_{-1}$ дуже мале. Визначимо $\alpha \equiv \frac{p-1}{1-p_{-1}} \approx (1 - p_{-1})^{-1} \gg 1$. Тоді наближенням для ймовірності хибнопозитивного результату є наступне

$$p_{FP} \approx p_{D \leftrightarrow D}{}^{\alpha^{-1}} + p_{U \to D} \tag{2.31}$$

Оскільки $p_{D \leftrightarrow D} \alpha^{-1} \leq \alpha^{-1}$, то серед позитивно помилково спрацьовуючих помилок переважає $p_{U \to D}$, коли $p_{U \to D} \gg \alpha^{-1}$. Крім того, зверніть увагу, що для гідної системи виявлення дублікатів $p_{D \leftrightarrow D}$, ймовірно, буде надзвичайно малим, оскільки система має широкі знання про всі свої оригінальні зображення. У цьому конкретному випадку здійснюється подальше наближення: $p_{FP} \approx p_{U \to D}$.

Далі ми припускаємо, що пріори однакові для всіх додатних c_t, а саме $p_t = (1 - p_{-1}) / |O|$. Іншими словами, тестове зображення має однакову ймовірність бути копією оригіналу чи іншого. У цьому випадку $p_{D \leftrightarrow D}$ і $p_{D \to U}$ спрощуються до

$$p_{D \leftrightarrow D} = \frac{1}{|O|} \sum_{i=1}^{|O|} \Pr\{c_e \neq c_t, c_e > 0 | c_t = i\} \tag{2.32}$$

$$p_{D \leftrightarrow U} = \frac{1}{|O|} \sum_{i=1}^{|O|} \Pr\{c_e = -1, c_e > 0 | c_t = i\} \tag{2.33}$$

та оцінки коефіцієнтів помилок для порогового значення селективності u, тоді задаються

$$\hat{p}_{D \to D}(u) = \frac{1}{|O| \cdot \sum_{j=1}^{|\mathcal{L}|} I_{\{x:x>0\}}(\mathcal{L}(j))} \sum_{\substack{i=1 \\ \text{за умови } \mathcal{L}(i)>0}}^{|T|} fp\left(\mathcal{L}(i), \mathrm{d}_{\mathrm{O}}^{\mathrm{N}}(T(i), u)\right) \tag{2.34}$$

$$\hat{p}_{D \to U}(u) = \frac{1}{|O| \cdot \sum_{j=1}^{|\mathcal{L}|} I_{\{x:x>0\}}(\mathcal{L}(j))} \sum_{\substack{i=1 \\ \text{за умови} \mathcal{L}(i)>0}}^{|T|} fn\left(\mathcal{L}(i), \mathrm{d}_{\mathrm{O}}^{\mathrm{N}}(T(i), u)\right) \tag{2.35}$$

$$\hat{p}_{U \to D}(u) = \frac{1}{\sum_{j=1}^{|\mathcal{L}|} I_{\{-1\}}(\mathcal{L}(j))} \sum_{\substack{i=1 \\ \text{за умови} \mathcal{L}(i)=-1}}^{|T|} fp\left(\mathcal{L}(i), \mathrm{d}_{\mathrm{O}}^{\mathrm{N}}(T(i), u)\right) \tag{2.36}$$

де T - набір тестових зображень, визначених у частині 4.3.1, L - відповідний набір міток, $\mathrm{d}_{\mathrm{O}}^{\mathrm{N}}$ (T (i), u) - функція (визначена у частині 4.2.2), яка

оцінює мітку класу тестового зображення T (i) щодо набору оригінальних зображень O,

fp (·, ·) та fn (·, ·) - це функції, які вказують на помилково позитивні та помилково негативні помилки, як визначено раніше в цій частині.

Для оцінки ефективності алгоритму, який виявляє дублікати кількох оригіналів, ми використовуємо одну систему, налаштовану на набір оригінальних зображень. Крива DET будується шляхом збору ймовірностей $\hat{p}_{FP}$ та $\hat{p}_{FN}$, оцінених з використанням ймовірностей $\hat{p}_{D\leftrightarrow D}(u)$, $\hat{p}_{D\to U}(u)$ та $\hat{p}_{U\to D}(u)$, обчислених для різних значень порогу u.

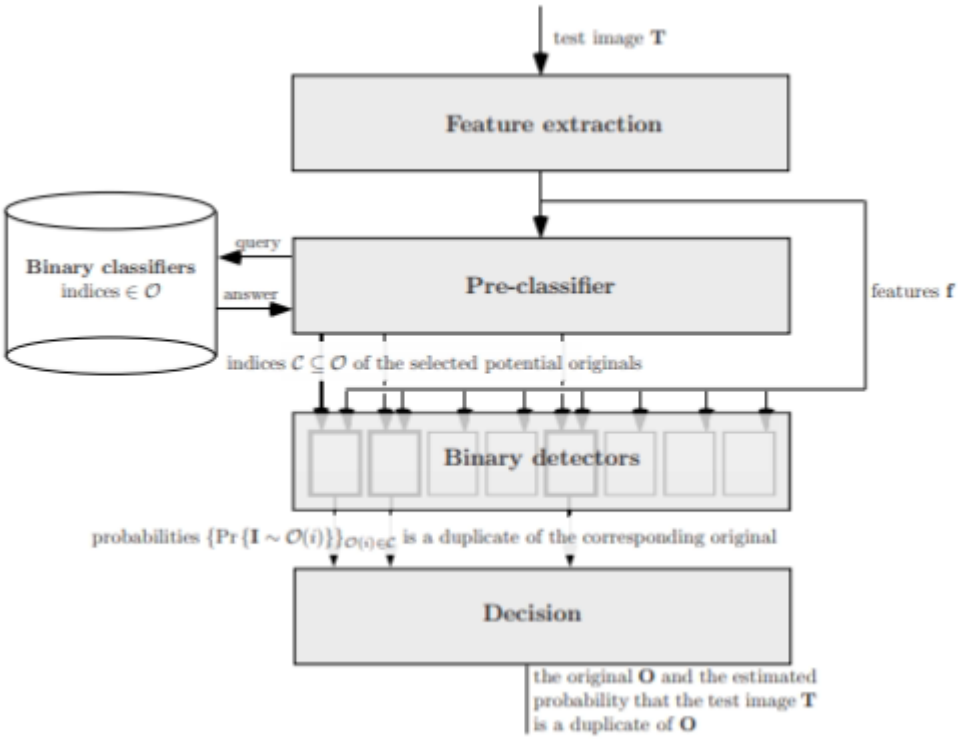


Рисунок 2.15 Блок-схема підходу виявлення дублікатів. Описано чотири основні кроки. Крок вилучення ознак витягує відповідні візуальні особливості f із тестового зображення T. Крок попереднього класифікатора вибирає потенційні оригінальні зображення C ⊆ O зі структури індексації. Останній індексує відповідні повторювані різновиди та використовує індекси серед множини O. Для кожного найбільш вірогідного вихідного зображення O (i) ∈ C бінарні детектори оцінюють ймовірності Pr {I ∼ O (i)}, що тестове зображення

T є дублікат оригінального зображення O (i). Нарешті, на етапі прийняття рішення або вирішується, що тестове зображення T не пов’язане з будь-яким із вибраних потенційних оригінальних зображень у C, або що O ∈ C є найбільш вірогідним оригіналом.

2.2.4 Огляд підходу та загальні компоненти

Підхід, запропонований у цьому дипломі, спирається на спостереження, зроблені в частині 2.2.1. Точніше, дублікати зображення лежать, за певних умов, на гладкому колекторі. Далі ми припускаємо, що колектори, що визначаються різними оригінальними зображеннями, відрізняються як за розміщенням, так і за формою. Це додаткове припущення призводить до підходу, коли різновиди явно оцінюються для кожного оригінального зображення. Наслідуючи цю ідею, спочатку пропонується ефективний метод, який оцінює ймовірність того, що тестове зображення лежить на цьому колекторі. Далі ми називаємо цей єдиний оригінальний детектор дублікатів зображення двійковим детектором. Потім цей підхід поширюється на випадок, коли доступно багато оригінальних зображень, і тестове зображення має бути ствердженим як копія одного з них або не пов’язане з яким-небудь із них. Цей розширений підхід до виявлення дублікатів частинаений на чотири основні етапи, які проілюстровано блок-схемою на рисунку 4.6. У частині 4.4.2 введено етап вилучення ознак. Решта кроків докладно описані у наступних двох главах. Точніше, крок бінарних детекторів детально описаний у главі 5, тоді як етап попереднього класифікатора та рішення детально підготовлені в главі 6.

Попередня обробка зображень

Цей частина описує операції попередньої обробки, які застосовуються до зображення перед вилученням об’єкта. Операції попередньої обробки мають дві цілі. По-перше, зображення описується у кольоровому просторі, що дозволяє легко отримувати важливу інформацію. По-друге, вводиться слабка

форма стійкості до таких перетворень, як зміна розміру, обрамлення або зміна інтенсивності. Далі ми вважаємо, що зображення висоти I і ширини J задається трьома матрицями I × J R, G та B, що відповідають червоному, зеленому та синьому каналам відповідно. Матриці індексуються наступним чином. $R_{(i, j)}$ відповідає елементу, поданому на i-му рядку j-го стовпця R. Так само $R_{(i, \cdot)}$ відповідає i-му рядку, а $R_{(\cdot, j)}$ до j-та колонки.

Перш за все, ми додаємо слабку стійкість до кадрування, видаляючи майже постійні рядки та стовпці. Видалення рядків і стовпців виконується ітеративним способом; одна ітерація описана нижче. Спочатку обчислюємо для кожного рядка і стандартне відхилення $\sigma_{(R, i)} = \text{std}(R_{(i, \cdot)})$ на червоному каналі, аналогічно $\sigma_{(G, i)}$ та $\sigma_{(B, i)}$ на зеленому та синьому каналах. Потім обчислюється усереднене стандартне відхилення σ_i для кожної лінії i та кожного кольорового каналу. Нарешті, рядок i видаляється лише тоді, коли відповідне усереднене стандартне відхилення σ_i менше, ніж $s \cdot \sum_i \sigma i / \tilde{J}$, де s - параметр, який керує чутливістю алгоритму видалення рядка, а $\tilde{J}$ - поточна кількість стовпців . Для першої ітерації $\tilde{J}$ дорівнює кількості стовпців на зображенні. Зазвичай для s встановлюється значення 0,1 в наступних. Подібним чином майже "постійні" стовпці видаляються. Процес остаточно повторюється, доки існують рядки або стовпці, які потрібно видалити.

Потім вводиться слабка форма незмінності масштабу шляхом зміни розміру зображення. Точніше, розмір зображення змінено таким чином, що воно містить приблизно 214 пікселів, це число відповідає квадратному зображенню 128 × 128 пікселів, зберігаючи вихідне співвідношення сторін. Окрім слабкої форми інваріантності масштабу, розмір попередньо обробленого зображення в основному є постійним, незалежно від розміру тестового зображення, це також дозволяє пришвидшити виділення функцій за рахунок зменшення кількості пікселів для обробки.

Потім масштабоване зображення представляється у модифікованому просторі інтенсивності насиченості відтінку (HSI): логарифмічному просторі

відтінків, насиченості та вирівняної інтенсивності. Більш конкретно, логарифмічний відтінок H^{log} визначається наступним чином [41]

$$H^{log} = \frac{\log R - \log G}{\log R + \log G - 2\log B} \tag{2.37}$$

де R, G і B - червоний, зелений і синій канали, і операції виконуються поелементно. Логарифмічний відтінок має перевагу в тому, що він незмінний до змін гами та яскравості. Насиченість S така ж, як і для класичного HSI [42], і дається

$$S = 1 - \frac{3 \cdot \min(R,G,B)}{R+G+B} \tag{2.38}$$

де операції застосовуються поелементно. За будовою насиченість досить інваріантна до змін освітленості. Нарешті, вирівняна інтенсивність I^{equ} задається формулою

$$I^{equ} = equ\,(\frac{R+G+B}{3}) \tag{2.39}$$

де equ (·) - глобальний оператор вирівнювання гістограм [42], а операції додавання та ділення виконуються поелементно. Вирівнювання дозволяє зробити інтенсивність переважно інваріантною до змін гами та яскравості, як показано в [Maret et al., 2006a].

name	feature type	number of features
Gabor	mean of the squared coefficients	30
Gabor	standard deviation of the squared coefficients	30
Colour	class histogram	10
Colour	mean of each class	24
Colour	standard deviation of each class	24
Colour	spatial distribution of each class	20
Grey-level	class histogram	8
Grey-level	spatial distribution of each class	16
		total = 162

Рисунок 2.16 Огляд використаних функцій. У цій таблиці перелічені використані функції, типи статистики та кількість вилучених значень.

Особливості

Ця частина представляє особливості, що використовуються для експериментів, проведених через інші глави. Зверніть увагу, що більш ретельне обговорення загального видобування візуальних ознак можна знайти в частині 2. Функції витягуються після обробки зображення, як описано в частині 4.4.1.

Особливості, використані в цьому дипломі, три типи: текстура, кольори та статистика рівня сірого. Вони подібні до тих, що використовуються в [3], в яких виявлено, що вони дають хороші результати у програмах виявлення дублікатів зображень. Основні відмінності - це додані 24 функції рівня сірого та відсутність „місцевої" статистики. Додані функції рівня сірого фіксують характеристику, пропущену кольоровими властивостями, а саме розподіл рівня інтенсивності, та забезпечують підвищену ефективність, як це продемонстровано в главі 5 та главі 6. У таблиці 4.3 узагальнено 162 використані функції. Потім кожна група ознак детально описана в наступних підчастинаах.

Текстура

Особливості текстури складаються із статистики першого та другого порядку кожного піддіапазону перетворення Габора. Останнє виконується, як у [44] на каналі зрівняльної інтенсивності I^{equ}. Для того, щоб бути автономним, зараз підсумовано конструкцію набору фільтрів Габора. Двовимірне перетворення Фур'є материнського фільтра Габора G (u, v) можна записати наступним чином

$$G(u,v) = \exp - \ 0.5(\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_u^2}) \qquad (2.40)$$

де u та v - горизонтальна та вертикальна частоти відповідно. Параметр W контролює центральну горизонтальну частоту, тоді як параметри σ_u та σ_v регулюють горизонтальну та вертикальну ширини фільтра відповідно. Потім можна отримати набір орієнтованих фільтрів Габора $\{G_{mn}(u, v)\}_{mn}$, виконавши зміну змінних на материнському фільтрі

$$G_{mn}(u,v) = a^{-m} \cdot G(a^{-m}u\cos\Theta + a^{-m}\ v\sin\Theta, -a^{-m}\ \ u\sin\Theta + a^{-m}v\cos\Theta)41$$

де θ = nπ / K, K - загальна кількість орієнтацій, a - коефіцієнт, більший за одиницю, a n і m - індекси орієнтації та шкали відповідно. Набір фільтрів Gabor досить зайвий, оскільки фільтри перекриваються. У роботі [45] ця надмірність зменшується шляхом вибору параметрів фільтрів (a, σ_u, σ_v та W) так, щоб контури півпікової величини сусідніх фільтрів торкалися, але не перекривались. Ці параметри контролюються чотирма метапараметрами: верхня центральна частота, що представляє інтерес U_h, нижня центральна частота, що цікавить U_l, кількість орієнтацій K та кількість шкал S. Потім параметри задаються

$$a = \left(\frac{U_h}{U_l}\right)^{-\frac{1}{s-1}} \tag{2.42}$$

$$\delta_u = \frac{(a-1)U_h}{(a+1)\sqrt{2ln2}} \tag{2.43}$$

$$\delta_v = \tan\left(\frac{\pi}{2K}\right)\left(U_h - 2\ln\left(\frac{\sigma_u^2}{U_h}\right)\right)\left(2ln2 - \frac{(2ln2)^2\sigma_u^2}{U_h^2}\right)^{-0.5} \tag{2.44}$$

і W = U_h, n = 0, 1,. . . , K - 1, a m = 0, 1,. . . , S - 1. Крім того, чутливість фільтрів до глобальних змін інтенсивності усувається додаванням констант до фільтрів Габора, щоб середні значення їх дійсних частин дорівнювали нулю.

У цій дипломній роботі фактично використовуються параметри U_h = 0,75 для верхньої центральної частоти, U_l = 0,05 для нижньої центральної частоти, п’ять шкал S = 5 та шість орієнтацій K = 6. В результаті отримують 30 фільтрів. Потім зображення фільтрується за допомогою цих 30 фільтрів, в результаті чого отримується 30 зображень піддіапазону. Отримані 30 зображень піддіапазону нарешті підсумовуються до оцінок середніх значень та стандартних відхилень їх квадратних коефіцієнтів. Це призводить до загальної оцінки 30 середніх та 30 дисперсій.

Колір

Колірні особливості обчислюються в кольоровому просторі HSI. Кожен піксель на зображенні класифікується в одному з десяти кольорових класів

залежно від його положення в цьому просторі. Класи - це ахроматичні кольори (S = 0) чорний, сірий та білий, а хроматичні кольори (S> 0) червоний, оранжевий, жовтий, зелений, блакитний, синій та фіолетовий. Вирівняна інтенсивність використовується для класифікації пікселя до одного з трьох ахроматичних класів. Логарифмічний відтінок використовується для класифікації пікселя в одному із семи хроматичних класів.

Спочатку обчислюється гістограма класів кольорів. Це дає пропорцію кожного класу кольорів на зображенні. Потім обчислюється статистика каналів. Для кожного класу кольорів, хроматичного або ахроматичного, обчислюються середні та дисперсійні оцінки вирівняного каналу інтенсивності. Остаточно обчислюється форма просторового розподілу кожного класу кольорів. Це досягається обчисленням двох характеристик фігур для кожного класу кольорів, а саме розповсюдженість і подовження [46;47]. Перша характеристика вимірює компактність просторового розподілу класу кольорів. Другий дає співвідношення між довжиною та шириною фігури.

Рівень сірого

Характеристики рівня сірого базуються на вирівняному каналі інтенсивності моделі HSI. Динамічний діапазон зображення лінійно частинаений на вісім контейнерів, що відповідають стільки ж класам. Кожен піксель зображення потрапляє в один із цих контейнерів.

Спочатку обчислюється гістограма класів сірого рівня. Він дає пропорцію пікселів зображення для восьми діапазонів інтенсивності. Він нормований таким чином, що підсумовує до одиниці і містить 8 значень. Подібно до кольору, нарешті обчислюється форма просторового розподілу кожного класу рівня сірого. Це призводить до 8 показників розповсюдження і 8 показників подовження.

2.2.5 Підсумок глави

У цій частині представлена структура для виявлення дублікатів зображень. Рамка виявлення дублікатів спочатку складається з моделі дублікатів. Ця модель дозволяє дослідити характеристики підпростору, охопленого дублікатами зображення; наприклад, встановлено, що за певних припущень цей підпростір є багатоманітністю, вбудованою в простір зображення. Другим елементом фреймворку є загальна система виявлення дублікатів. Завдяки цій загальній системі ми розробляємо наш погляд на виявлення дублікатів, а саме на класифікацію тестового зображення в один із класів K + 1. K-класи відповідають K-оригінальним зображенням, відомим системі, або іншими словами "тестове зображення є дублікатом одного з відомих оригіналів", тоді як решта класів означає "тестове зображення не пов'язане ні з одним із відомих оригіналів зображення ". Нарешті, останній елемент системи стосується методології оцінки системи виявлення дублікатів, що базується на класифікаційному підході.

3 ЄДИНА ОРИГІНАЛЬНА СИСТЕМА ВИЯВЛЕННЯ ДУБЛІКАТІВ

У цій частині ми докладно описуємо наш підхід до виявлення дублікатів зображення одного оригінального зображення. Підхід частково базується на попередніх роботах [48]. Основна ідея запропонованої системи - адаптувати виявлення дублікатів до конкретного вихідного зображення. Потім система може класифікувати тестові зображення як дублікати оригінального зображення або як не пов’язані між собою зображення. Огляд системи вперше наведено в частині 5.1. Приклад навчання, а також показник ефективності, який використовується для побудови детекторів, наведено в частині 5.2. Потім система детально описана в частині 5.3. Потім продуктивність системи та аналіз детально описані в частині 5.4. Нарешті, можливі напрямки досліджень запропоновані в частині 5.5.

3.1 Огляд системи

Зараз ми представляємо огляд запропонованої єдиної оригінальної системи виявлення дублікатів зображень або двійкового детектора. Система складається з чотирьох етапів, як показано на малюнку 5.1, кожен з яких окреслений далі. Перш ніж йти далі, зауважте, що метод можна розкласти на дві різні частини. Перший, що складається з кроку, показаного у верхній частині рисунка 5.1, не залежить від оригінального зображення. І навпаки, другий, що включає кроки, показані в нижній частині рисунка 5.1, залежить від оригінального зображення; тому необхідне навчання. Використані навчальні приклади та показники представлені в частині 5.2, тоді як додаткові подробиці про залежні від кроку навчальні процедури наведені, а також детальний опис кожного етапу в частині 5.3.

Вилучення особливостей

Метою етапу вилучення об’єкта є відображення зображень у загальний простір, де порівняння є більш ефективним. Для цього з тестового зображення витягують глобальну статистику, таку як кольорові канали та текстури. Тестове зображення T спочатку обробляється, як описано в частині 3.4.1. Потім витягуються функції для попередньо обробленого зображення. Більш детально перелік функцій описаний у частині 4.4.2. Цей другий крок призводить до отримання вектора f, що містить D елементи.

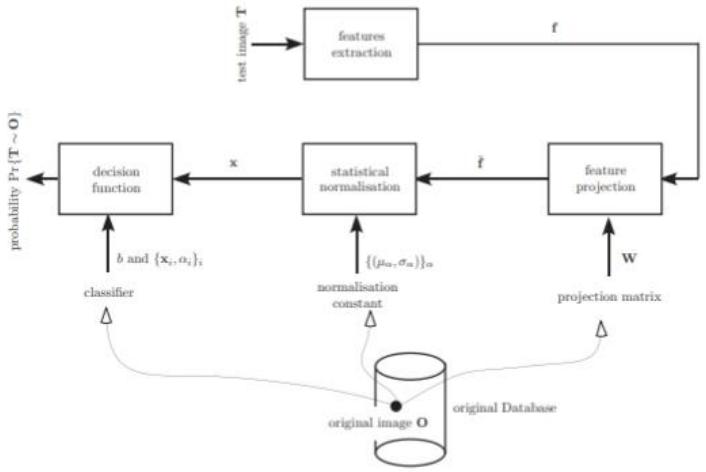


Рисунок 3.1: Структурна схема двійкового детектора. Системі передається тестове зображення, яке визначає, чи є воно дублікатом оригінального зображення O, для якого побудований детектор. Метод можна розкласти на дві окремі частини: етап, який не залежить від оригінального зображення, верхню частину малюнка та етапи, які залежать від оригінального зображення, нижню частину малюнка.

Проекція функції

На другому етапі функції трансформуються лінійно, щоб отримати кращий частина між дублікатами вихідного зображення та не пов’язаних між собою зображень. Точніше, прогнозовані ознаки даються $\tilde{\mathbf{f}} = \mathbf{W} \cdot \mathbf{f}$, де **W** - матриця проекції D × D. Проекція може залежати від оригінального зображення або бути спільною для кожного оригіналу. Далі **W** знайдено за допомогою

простого алгоритму аналізу основних компонентів (PCA) і є загальним для кожного оригіналу.

Статистична нормалізація

На третьому етапі елементи $\tilde{\mathbf{f}}$ нормуються щодо статистичного розподілу дублікатів. Відповідно, параметризація цього кроку залежить від оригінального зображення. Метою цього кроку є надання однакової важливості кожній ознаці, незалежно від діапазону їх значень. Це призводить до нормованого вектора x, що містить D елементи.

Функція прийняття рішень

На останньому кроці для визначення ймовірності Pr {**T** ~ **O**} використовується нелінійна функція прийняття рішення, що тестове зображення **T**, представлене шаблоном **x**, є дублікатом оригінального зображення **O**. Очевидно, цей крок параметризований відповідно до вихідного зображення.

3.1.2 Зауваження щодо навчання

Як уже згадувалося раніше, останні три кроки, показані на рисунку 3.1, повинні бути параметризовані відповідно до вихідного зображення і, отже, вимагають навчання. У цій частині ми представляємо навчальну процедуру, яка виконується каскадно. По-перше, обчислюється матриця проекції W, а потім прогнозовані ознаки нормалізуються. І нарешті, функція прийняття рішення тренується з використанням нормованих ознак.

Решта цього частинау складається з двох частин. У першій частині представлені приклади, що використовуються для навчання системи, тоді як у другій частині розглядаються показники, що використовуються для оцінки навчальних результатів.

У цій роботі дублікати генеруються за допомогою операцій, перелічених у таблиці 5.1. Зверніть увагу, що для створення детекторів, які добре працюють

як з тестами Qamra, так і з StirMark, використовується один навчальний набір, див. Частина 4.3 для отримання додаткової інформації про критерії. Вибір цих конкретних прикладів навчання зараз мотивований. Спочатку ми визначаємо перетворення, які в середньому призводять до векторів ознак, що знаходяться далі від оригіналу. Порядок перетворень, як і в попередньому розумінні, експериментально виявляється таким

1. обертання та обертання / масштабування;
2. Стиснення та обрізання JPEG;
3. зміни насиченості та зміни інтенсивності;
4. забарвлення;
5. зміни співвідношення сторін та зменшення вибірки;
6. масштабування та деформація;
7. лінійне перетворення та частотний режим видалення Лапласа.

Потім ми призначаємо кількість дублікатів на кожне перетворення, пропорційне відповідній середній відстані. Після, діапазон параметризації кожного перетворення рівномірно відбирається. Нарешті, ми додали шість дублікатів з наступних непараметризуваних перетворень до навчального набору

1. зміни контрасту (плюс і мінус);
2. видалення плям;
3. зменшення глибини кольору;
4. горизонтальне відображення;
5. перетворення рівня сірого.

Зверніть увагу, що ця остання трансформація, перетворення рівня сірого, не є частиною тестів Qamra та StirMark. Однак це додано, щоб детектори добре працювали для тестових зображень рівня сірого. Таким чином, це дає 106 позитивних прикладів, наведених у таблиці 5.1.

Нарешті, важливо зауважити, що оптимальний вибір прикладів навчання все ще залишається відкритим питанням і є центром подальших досліджень. Однак деякі можливі вказівки наведені в частині 3.5.

Навчальна метрика

Показник F-оцінки використовується для оцінки ефективності виявлення на етапі навчання. F-оцінка визначається наступним чином [49]

$$F(TP, FP, P) = \frac{TP}{P} \times \frac{TP}{TP+FP} \quad (3.1)$$

де P - загальна кількість позитивних випадків, TP - кількість позитивних випадків, правильно класифікованих, а FP - кількість негативних випадків, неправильно класифікованих. Перший доданок у правій частині рівняння (5.1) відповідає виклику. І навпаки, другий доданок представляє точність. F-оцінка врівноважує ці дві суперечливі властивості: точність зростає із зменшенням кількості помилкових спрацьовувань, а повнота зменшується із зменшенням кількості помилково негативних результатів, як правило, це означає, що кількість помилкових спрацьовувань збільшується. Рівняння рівняння (3.1) можна переписати як

$$F_p(\hat{p}_{FP}, \hat{p}_{FN}) = (1 - \hat{p}_{FN}) \times \frac{(1-\hat{p}_{FN})}{1 \cdot \hat{p}_{FP} - \hat{p}_{FN}} \quad (3.2)$$

де $\hat{p}_{FP}$ = FP / N та $\hat{p}_{FN}$ = FN / P - оцінені коефіцієнти помилкових позитивних спрацьовувань та частота помилок з негативними значеннями, як визначено у частині 4.3. ρ = N / P дає співвідношення між кількістю негативних і позитивних випадків. Що стосується рівняння (5.1), то перший член у правій частині рівняння (5.2) відповідає виклику, а другий точності. В решті документа ми використовуємо формулювання, подане рівнянням (5.2). Один недолік цієї метрики полягає у відношенні ρ між кількістю негативних та позитивних випадків; це потрібно знати заздалегідь.

3.1.3 Бінарний детектор

Тепер ми детально описуємо запропоновану єдину оригінальну систему виявлення дублікатів. Зокрема, кроки, представлені в нижній частині рисунку 5.1, а саме проектування, нормалізація та функція прийняття рішень, деталізуються разом із навчальними процедурами, коли це потрібно. З іншого боку, етап, представлений у верхній частині рисунка 5.1, а саме вилучення ознак, вже описаний у частині 4.4.

categories	♯	parameterisations
Colourising	18	Tint the red, green, or blue channel from -11% to +11% by steps of 4%
Contrast changes	2	Increase or decrease the contrast[a]
Despeckling	1	Apply ImageMagick's despeckling operation
Downsampling	5	Downsample by 3% to 83% by steps of 20%
Colour depth reduction	1	Reduce the colour palette to 256 colours
Saturation changes	8	Change the values of the saturation channel by -22% to +22% by steps of 6%
Intensity changes	8	Change the intensity with the same parameters than for saturation
JPEG compression	10	JPEG compression with quality factors from 8 to 98 by steps of 10
Shearing	4	shearing in (X°, Y°) directions with X and Y varying from 0° to 6° by steps of 3°
Cropping	10	centred cropping from 2% to 94% by steps of 10%
Flipping	1	horizontal flip
Scaling	4	scaling by factors from 0.45 to 0.95 by steps of 0.2
Aspect ratio	5	change aspect ratio of X(Y) by a factor from 0.75 to 1.25 by steps of 0.2
Rotation	11	rotations by angles from 4° to 92° by steps of 8°
Rotation/scaling	11	same as above but followed by scaling
Linear transform	3	general linear geometric transformation $\mathbf{c}' = \mathbf{Tc}$ [b]
FMLR	3	frequency mode Laplacian removal attack with parameters 0.02, 0.04, and 0.06
Grey-level conversion	1	
total	106	

[a] using ImageMagick's [Still, 2005] default parameter
[b] same matrices as for testing but with entries multiplied by 0.99.

Рисунок 3.2 Навчальні дублікати. Встановлено, що ці дублікати дають початок детекторам дублікатів, які добре працюють на еталонах Qamra та StirMark.

Особливості проекції

Ідея цього кроку полягає у спроектуванні об’єктів у простір, що дозволяє частинаяти дублікати свердловин та недублікатів. Перетворення може бути лінійним або нелінійним, і, крім того, воно може бути адаптоване до початкового або незалежне від нього. Подальші експерименти, проведені в рамках цієї дипломної роботи, показали, що кращі результати отримуються простим використанням PCA на великому наборі зображень для отримання проекційної матриці W, загальної для всіх детекторів.

Точніше, алгоритм PCA проектує особливості, знаходячи напрямки, по яких максимізується розкид хмари точок [50]. Іншими словами, прогнози PCA призводять до хорошого представлення даних. Обчислюючи PCA на елементах, що представляють різні зображення, ми таким чином отримуємо проекцію, яка добре відокремлює зображення, у тому сенсі, який наводився раніше, зображень. З цієї причини PCA використовується на зображеннях, не пов’язаних із вихідним зображенням, щоб знайти проекційну матрицю D × D, W, загальну для всіх детекторів. А прогнозовані особливості задаються

$$\tilde{f} = W \cdot f \tag{3.3}$$

Зверніть увагу, що матриця W залежить від характеристики колекції зображень, але не залежить від вихідного зображення. З цієї причини його можна обчислити на дуже великому наборі функціональних векторів, і ту саму матрицю можна використовувати для кожного детектора.

Нормалізація

Мета нормалізації полягає в тому, щоб забезпечити співмірність елементів ознак або, іншими словами, щоб діапазон записів векторів ознак був порівнянним. Прогнозовані ознаки $\tilde{f}$ нормуються з використанням методу статистичної нормалізації [51]. Точніше кажучи, нехай μ_α та σ_α є середнім та стандартним відхиленням оцінок α-го прогнозованої ознаки для підмножини навчального набору. Точніше, навчальна підмножина складається з дублікатів

прикладів оригіналу, і, щоб уникнути врахування викидів, приклади тренувань, для яких будь-яка особливість є екстремумом порівняно з навчальним набором, ігноруються. Тоді нормована α-та ознака x_α визначається як

$$x_\propto = \frac{\tilde{f}_\propto - \mu_\alpha}{k \cdot \sigma_\alpha} \tag{3.4}$$

де $\tilde{f}_\alpha$ - прогнозована ознака, наведена у рівнянні (3.3). За теоремою Чебичева, принаймні частка 1 - 1 / k^2 від $\tilde{f}_\alpha$ знаходиться в інтервалі [−1, 1]. У наступному k встановлено значення 3, так що більше 90% функцій x_α знаходяться в межах [−1, 1].

Функція прийняття рішень

Функція прийняття рішення повинна визначити, чи відповідає вектор x дублікату вихідного зображення. Це проблема двійкової класифікації, де два класи відповідають дублікатам та недублікатам відповідно. Мета полягає у створенні, використовуючи обмежену кількість навчальних прикладів, класифікатора, який добре узагальнює нові моделі. Для цього можна використовувати багато алгоритмів класифікації. Встановлено, що ефективність SVC перевершує ці два підходи. Можливі причини такі. SVDD створює дуже жорсткі межі і, отже, є більш схильним до надмірних тренувань. З тієї ж причини SVDD є більш чутливим до обраних прикладів навчання, ніж SVC. З іншого боку, підхід прамокутника забезпечує лише дуже грубе наближення і, отже, призводить до гірших показників, ніж SVC. Далі ми використовуємо лише функцію прийняття рішень на основі SVC.

Основні SVC [52;53] - це двійковий класифікатор, який частинаяє два класи гіперплощиною. Крім того, нелінійні ядра дозволяють відображення шаблонів у простір, де їх можна краще розрізнити гіперплощиною. Більше інформації про SVC можна знайти на малюнку 2.3.4. Далі ми спочатку даємо короткий огляд ν-SVC перед деталізацією вибору навчальної процедури.

Огляд ν-SVC

Ми використовуємо ν-параметризацію [54;55] з SVC, і радіальна основа функціонує як ядро. Подвійна обмежена задача оптимізації подана у рівнянні (5.5). У подвійній формі лагранжіан максимізується за рахунок оптимізації мультиплікаторів Лагранжа α_i

$$\max_{\propto} -\frac{1}{2}\sum_{i,j=1}^{m} \propto_i \propto_j y_i y_j \ker(x_i, x_j) \tag{3.5}$$

з урахуванням обмежень $\sum_{i=1}^{m} \alpha_i y_i = 0$, $\sum_{i=1}^{m} \alpha_i \geq \nu$, і $0 \leq \alpha_i \leq 1 / m$, де m - кількість прикладів тренувань, x_i - схеми тренувань, y_i - відповідні тренування мітки (-1 для негативних прикладів та +1 для позитивних прикладів), а ker $(\cdot, \cdot)$ - це функція ядра, що задовольняє умовам Mercer. У цій роботі ми використовуємо ядро з радіальною базовою функцією (RBF), задане

$$\ker(x_i, x_j) = \exp(-y \cdot |x_i - x_j|^2) \tag{3.6}$$

Нарешті, ядро RBF регулюється лише одним параметром замість двох для ядра поліномів.

Параметри цього методу класифікації: $\nu \in [0, 1]$ та $\gamma \in R^+$. Параметр ядра γ контролює складність межі рішення. Обмежену задачу оптимізації, наведену в рівнянні (5.5), можна вирішити за допомогою стандартних методів квадратичного програмування.

Нарешті, функція прийняття рішення вказує, до якого класу належить тестовий шаблон z. Це дано

$$f(z) = sgn\left(\sum_{i=1}^{m} y_i \propto_i \ker(z, x_i) + b\right) \tag{3.7}$$

де константа b визначається векторами опори. Точніше, b задано як

$$b = y_k - \sum_{i=1}^{m} y_i \propto_i \ker(x_i, x_k) \tag{3.8}$$

для всіх x_k таких, що $0 < \alpha_k < 1 / м$. Значення $f = \sum_{i=1}^{m} y_i \alpha_i \ker(z, x_i) + b$ у рівнянні (5.7) називається кордоном і надає відстань до межі рішення.

Класифікатор вектора підтримки передбачає лише мітку класу, але не ймовірність бути цим класом. Далі ми коротко опишемо, як розширюється SVC для оцінки ймовірності. Детальніше у [56;57]. Враховуючи два класи даних, метою є оцінка для будь-якого візерунка x задньої ймовірності p_{+1} та p_{-1}, а саме

$$p_{+1} = \Pr\{y = +1|x\} \text{ і } p_{-1} = \Pr\{y = -1|x\} \quad (3.9)$$

Один із способів перетворення вихідного коду SVC на ймовірність полягає у підготовці безпосередньо класифікатора з використанням ядра на основі максимальної ймовірності. Більш відповідний метод запропонований Платтом, де сигмовидна функція відображає поля в оцінці ймовірності [Platt, 2000]. Перевага цієї методики полягає в тому, що задні ймовірності Pr {y = +1 | x} та Pr {y = −1 | x} отримуються безпосередньо, а умовну ймовірність класу оцінювати не потрібно. Сигмоїда задана

$$\Pr\{y = +1|x\} = \frac{1}{1+\exp(a \cdot f + b)} \quad (3.10)$$

де параметри a та b оцінюються шляхом мінімізації негативної функції логарифмічної вірогідності з використанням відомих навчальних даних та їх граничні значення f. Етикетки та значення рішень повинні бути незалежними, тому k-кратна перехресна перевірка може бути використана для отримання значень рішення [58].

3.1.4 Результати

У цій частині ми представляємо експериментальні результати для того, щоб оцінити ефективність запропонованого одиночного детектора дублікатів оригінального зображення. Перший експеримент, представлений у частині 3.4.1, порівнює ефективність запропонованої системи виявлення дублікатів із системою, заснованою на стандартних показниках, в даному випадку L_1 та L_2. Другий експеримент, описаний у частині 3.4.2, досліджує вплив параметризації

F-оцінки. Третій експеримент, зображений у частині 3.4.3, аналізує характеристики індивідуального детектора. Четвертий експеримент, розглянутий у частині 3.4.4, представляє простір для зберігання та обчислювальний ресурс, необхідний запропонованій системі виявлення дублікатів. Останні експерименти, описані в частині 3.4.5, аналізують ефективність запропонованої системи щодо двох інших сучасних методів.

3. 1.5 Порівняння з існуючими методами виявлення дублікатів

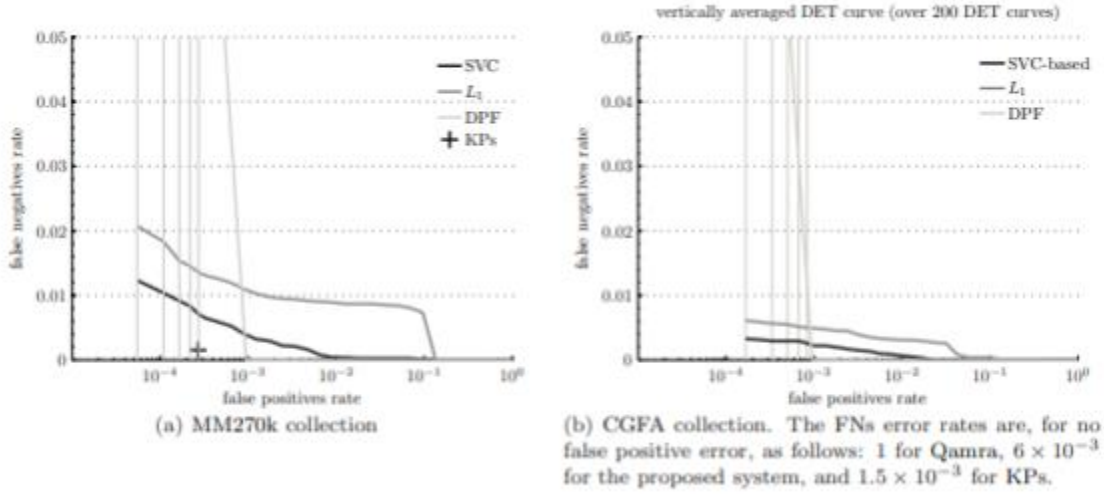
Тепер ми порівняємо ефективність запропонованого методу з ефективністю існуючих систем виявлення дублікатів. Вибір цих систем є непростим, оскільки, як зазначається в главі 3, не існує стандартизованого еталону, а методологія тестування часто не дається чітко. З цієї причини обираються системи, для яких набір чітко визначений, а саме [2] та [59]. Очевидно, ми маємо на увазі, що надається або колекція зображень, або дані трансформацій, що використовуються для визначення дублікатів, разом із їх параметрами. Наприклад, Ke et al. зробили доступними свої колекції зображень, а саме MM270k та CGFA, і використовують трансформації, запропоновані в Qamra et al., а саме Qamra. На жаль, жодна з цих двох публікацій не дає результатів для еталону StirMark, а отже, порівняння стосуються лише еталону Qamra.

Порівняння - результати та аналіз

Рисунок 3.10 порівнює ефективність запропонованої системи виявлення дублікатів із сучасними техніками, про які повідомляється в [2;60]. Чорна лінія відповідає усередненій по вертикалі кривій DET, отриманій у нашій системі. Світло-сіра лінія представляє приблизну ефективність методу виявлення дублікатів на основі функції сприймання відстані (DPF) [61]. Хрестик вказує на ефективність системи виявлення дублікатів на основі ключових моментів (KPs) [62].

Тепер, порівнюючи ефективність методу DPF з ефективністю запропонованої системи, можна спостерігати дві речі. По-перше, метод DPF не досягає помилки FNs для хибнопозитивних показників помилок вище 1×10^{-3}. Однак, опинившись нижче цієї точки, продуктивність надзвичайно швидко погіршується. Нагадаємо, що метод DPF складається з метрики, де для обчислення відстані між ними використовуються лише найбільш подібні записи у двох векторах. Хоча це покращує ймовірність того, що дублікати наближаються до оригіналу, це аналогічно збільшує ймовірність того, що непов'язані зображення наближаються.

З іншого боку, запропонований метод перевершує KPs. Дійсно, на колекції CGFA, KPs досягає коефіцієнта помилок FNs $1{,}5 \times 10^{-3}$ без жодної помилково позитивної помилки. З іншого боку, запропонований метод досягає для того самого набору випробувань коефіцієнта помилок помилково негативних результатів 6×10^{-3}, а також відсутність помилково позитивних помилок. Крім того, розрив у продуктивності дещо більший для колекції MM270k, але ці результати менш значущі, оскільки вони екстраполюються для методу KPs і, можливо, передбачувана ефективність KPs завищена. Однак, оскільки наближення досить ретельне, це означає, що отриманий дублікат детектора може працювати досить погано.



(a) MM270k collection

(b) CGFA collection. The FNs error rates are, for no false positive error, as follows: 1 for Qamra, 6×10^{-3} for the proposed system, and 1.5×10^{-3} for KPs.

Рисунок 3.3 Порівняння з сучасними методами. Запропонована система порівнюється з двома сучасними системами, а саме KPs [63] та DPF [64]. Тест проводиться на двох різних колекціях зображень, MM270k та CGFA, а тест Qamra використовується для створення дублікатів тесту. Щоб зберегти точку порівняння, також подано продуктивність системи L_1.

3. 1.6 Розвідувальні роботи

У цій частині ми представляємо можливі напрямки досліджень, пов’язані з темою цього частинау. Більша частина пропозиції стосується підвищення ефективності. Крім того, результати даються щоразу, коли проводились попередні експерименти.

Оптимальні приклади навчання

Вибір навчальних прикладів є важливою темою, яка не повністю розглянута в цій дипломній роботі. У цій частині ми проводимо експеримент, а саме майже подвоюємо кількість повторюваних прикладів шляхом більш тонкої вибірки перетворюваних параметрів перетворень. Отримані повторювані приклади навчання наведені в таблиці 5.3. Порівняйте це з прикладами, наведеними раніше в таблиці 5.1. Цей новий навчальний набір містить 200 дублікатів замість 106 раніше.

Малюнок 3.11 показує ефективність попереднього навчального набору та нового. Крім того, Малюнок 5.11b демонструє, що використання нових прикладів навчання зменшує більш ніж на п'ятдесят відсотків частоту помилок FNs у всьому діапазоні частот помилок FPs.

Цей попередній експеримент відкриває дуже цікавий напрямок досліджень для виявлення дублікатів, а саме вибір оптимального набору прикладів навчання. Можливе рішення випливає з розгляду дублікатів моделі, розроблених у частині 4.1. Дійсно, хороші навчальні приклади повинні якомога

більше визначати підпростір, який охоплюють дублікати. Отже, хороший навчальний набір - це той, який максимально рівномірно відбирає цей підпростір. Явно виражене формулювання для цієї вибірки, безумовно, є дуже складним, оскільки підпростір в найпростішому випадку являє собою різноманіття. Однак першим підходом може бути рівномірна вибірка кривої, визначеної одним перетворенням. Це набагато простіше, оскільки довжину кривої можна легко наблизити. Тоді інше питання полягає в тому, як діяти, коли перетворення контролюється більш ніж одним параметром? Нарешті, ще одне цікаве питання полягає в тому, чи залежить оптимальний навчальний набір від зображення, чи можна визначити один комплект, адекватний більшості оригінальних зображень.

categories	♯	parameterisations
Colourising	36	Tint the red, green, or blue channel from -11% to +11% by steps of 2%
Contrast changes	2	Increase or decrease the contrast[a]
Despeckling	1	Apply ImageMagick's despeckling operation
Downsampling	10	Downsample by 3% to 93% by steps of 10%
Colour depth reduction	1	Reduce the colour palette to 256 colours
Saturation changes	15	Change the values of the saturation channel by -22% to +22% by steps of 3%
Intensity changes	15	Change the intensity with the same parameters than for saturation
JPEG compression	19	JPEG compression with quality factors from 8 to 98 by steps of 5
Shearing	9	shearing in (X, Y) directions with X and Y varying from 0 to 6 by steps of 2
Cropping	19	centred cropping from 2% to 94% by steps of 5%
Flipping	1	horizontal flip
Scaling	8	scaling by factors from 0.45 to 0.95 by steps of 0.1
Aspect ratio	10	change aspect ratio of X(Y) by a factor from 0.75 to 1.25 by steps of 0.1
Rotation	22	rotations by angles from 4° to 92° by steps of 4°
Rotation/scaling	22	same as above but followed by scaling
Linear transform	6	general linear geometric transformation $\mathbf{c}' = \mathbf{T}\mathbf{c}$ [b]
FMLR	3	frequency mode Laplacian removal attack with parameters 0.02, 0.04, and 0.06
Grey-level conversion	1	
total	200	

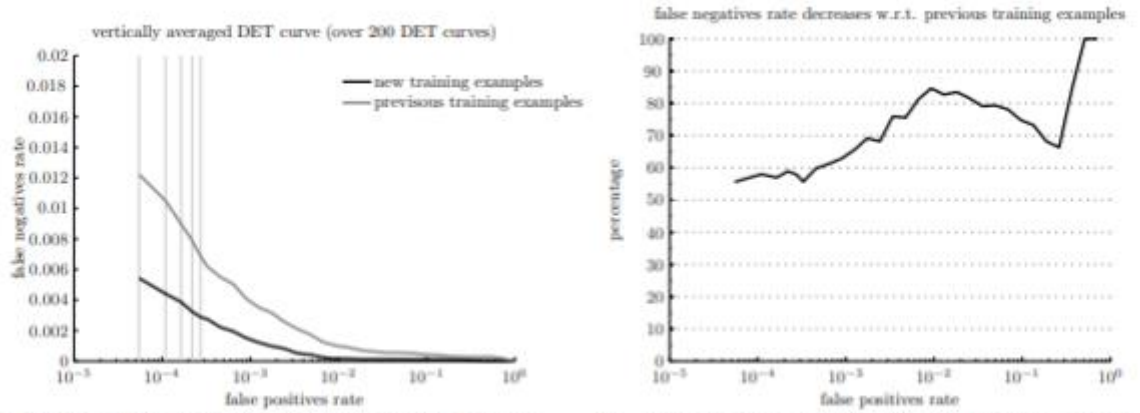
[a]using ImageMagick's [Still, 2005] default parameter
[b]same matrices as for testing but with entries multiplied by 0.99 and 1.01.

Рисунок 3.4 Нові приклади копій для навчання. Цей навчальний набір такий самий, як той, що наведений у таблиці 5.1, але параметризовані перетворення отримують більш точну вибірку. Він містить 200 дублікатів замість 116 раніше.

Поєднання класифікаторів

Ця пропозиція базується на результатах, показаних на рисунку 5.9. Можна помітити, що в деяких випадках не пов’язані між собою зображення, яким детектор призначає високі ймовірності, мають тони та колір, подібні до

відповідного оригіналу, але в різних кількостях. В ідеалі цього не повинно статися, оскільки вся необхідна інформація присутня в характеристиках, що описують зображення. Однак SVC не враховує цього факту. Це, найімовірніше, трапляється через обмежену кількість прикладів тренувань, наприклад, відсутність не пов’язаних між собою тренувальних зображень, які поводяться, як згадано вище.



(a) performance for the previous and the new training examples

(b) improvement brought by additional training examples

Рисунок 3.5 Покращення результативності завдяки більшій кількості прикладів навчання. Попередні навчальні приклади навчання наведені в таблиці 5.1, а нові навчальні приклади - у таблиці 5.3.

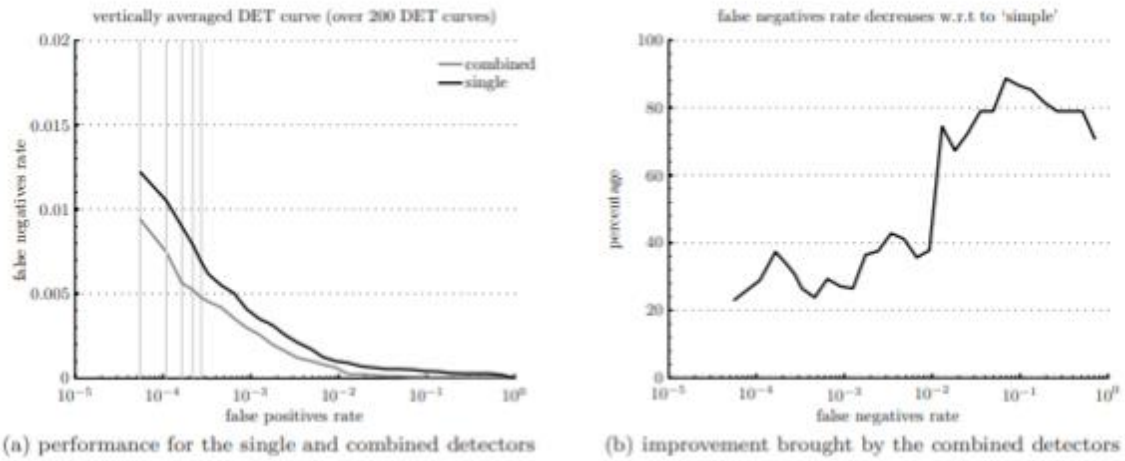
Можливий спосіб усунути цю ваду - включити більше прикладів навчання. Іншим можливим рішенням є створення іншого класифікатора, який просто враховує кількість кожного кольору на зображенні. Ми спробували цей підхід і витягли з векторів функцій записи, які представляють кількість кожного кольору та рівня сірого, що містяться на зображенні. Використовуючи точно ту саму процедуру, що і для всього вектора функцій, таким чином створюється дублікат детектора. Поодинці цей дублікат детектора працює досить погано. У поєднанні з детектором, використовуючи весь вектор, продуктивність значно покращується. Щоб проілюструвати цей момент, давайте розглянемо продуктивність, отриману для колекції зображень MM270k та для дублікатів,

сформованих за допомогою тестів Qamra та StirMark. Вони зображені на малюнку 5.12. Обидва детектори виводять імовірність виявлення, яка поєднується наступним чином

$$p_c = 1 - \frac{\sqrt{(1-p_1)^2 + w \cdot (1-p_2)^2}}{\sqrt{1+w}} \tag{3.11}$$

де $p_{1,2}$ - ймовірності, задані одиничними детекторами, а w - додатне число, яке використовується, щоб надати більше ваг одному з детекторів. Зверніть увагу, що ця комбінація насправді обернено пропорційна відстані до точки (1, 1). В експериментах w = 1/2, p_1 відповідає детектору з використанням усіх характеристик, тоді як p_2 - для більш простого детектора. Конкретний вибір w був мотивований тим, що p_1 є результатом більш ефективного детектора, ніж p_2. Малюнок 5.12a показує ефективність одиночного детектора та комбінацію. Крім того, на малюнку 5.12b показано, що комбінація зменшується приблизно на тридцять відсотків коефіцієнтів FNs нижче 1×10^{-2} і до восьмидесяти відсотків вище цього порогу. Дійсно, для частот FPs вище 1×10^{-2} комбінований детектор здатний майже не досягати помилок FNs.

Цей попередній експеримент відкриває дуже цікавий напрямок досліджень для виявлення дублікатів, а саме - поєднання класифікаторів [65].



(a) performance for the single and combined detectors (b) improvement brought by the combined detectors

Рисунок 3.6 Покращення продуктивності завдяки комбінації детекторів. Одиночний детектор використовує всі доступні функції, а потім поєднується з

більш простим детектором, який використовує лише функції, пов’язані з кількістю кожного кольору та рівнем сірого, представлених на зображенні.

3.1.7 Підсумок глави

У цій частині ми представили систему виявлення дублікатів на основі класифікатора опорних векторів (SVC). Потім ефективність запропонованої системи аналізується та порівнюється із сучасними методами. Нарешті, вивчаються можливі напрямки досліджень.

Запропонована система складається з чотирьох етапів, викладених далі. На першому кроці, описаному в попередньому частині, для опису зображення використовується глобальна статистика. На другому етапі функції трансформуються лінійно, щоб отримати кращий частина між дублікатами оригінального зображення та не пов’язаних між собою зображень. На третьому етапі елементи проектованої ознаки нормуються відповідно до статистичного розподілу дублікатів. На останньому кроці використовується нелінійна функція прийняття рішення, заснована на SVC, для визначення ймовірності того, що тестове зображення є дублікатом оригінального зображення.

Ефективність запропонованої системи оцінюється за допомогою стандартних орієнтирів та аналізується результат. Встановлено, що запропонований детектор-дублікат на основі SVC значно перевершує детектори, що використовують ті самі характеристики, але на основі метрики L_1. Крім того, система порівнюється з існуючими сучасними методами. Точніше, він перевершує метод DPF, який використовує більше можливостей для опису зображення. Хоча дещо перевершує метод KPs, запропонований метод у п’ять-десять разів менш обчислювально складний.

Нарешті, ефективність запропонованої системи можна значно покращити, використовуючи кращий приклад навчання або комбінуючи простіші класифікатори. Однак ці два напрямки досліджень вимагають подальших робіт.

3.2 Система виявлення копій кількох оригінальних зображень

У цій частині я докладно описую підхід до виявлення повторюваних зображень кількох оригінальних зображень. Підхід частково базується на попередніх роботах [66]. Основна ідея запропонованої системи полягає у створенні двійкового детектора дублікатів, як було розроблено раніше в главі 5, а потім у ефективному їх поєднанні. Потім система може класифікувати тестове зображення як дублікат одного з оригінальних зображень або як не пов'язане зображення. Основним внеском цієї глави є попередній класифікатор, запропонований для обрізання відомих системі зображень, що дозволяє уникнути використання кожного двійкового детектора з кожним тестовим зображенням.

Цей підхід в першу чергу мотивований у частині 6.1. Потім огляд системи наведено в частині 6.2. Деякі зауваження щодо навчання наведені в частині 6.3. Потім алгоритм попереднього класифікатора детально описаний у частині 3.2.4, а відповідні результати - у частині 6.5. Аналіз роботи всієї системи наведено в частині 6.6. Можливі напрямки досліджень нарешті запропоновані в частині 6.7.

3.2.1 Мотивація підходу

Система, представлена в цій главі, має на меті виявити дублікати одного з багатьох оригінальних зображень, відомих системі, тоді як система, представлена раніше, в главі 3.2, знала лише один оригінал. Для отримання додаткової інформації про відмінності між двома підходами читач повертається до частинів 3.1 та 3.2.

Основна ідея запропонованої системи виявлення дублікатів декількох оригінальних зображень полягає у використанні двійкових детекторів,

розроблених у частині 3.2, для кожного оригінального зображення. Потім їх комбінація визначає, чи є тестове зображення дублікатом одного з оригінальних зображень, відомих системі. Зауважте, що кількість оригіналів може бути досить великою залежно від програми, наприклад, у тисячах або навіть мільйонах. Тоді проблема полягає в наступному. При використанні набору двійкових детекторів, кожен з яких налаштований на певне оригінальне зображення, тестове зображення потрібно послідовно перевіряти з кожним двійковим детектором. На жаль, ця процедура стає сильно громіздкою, оскільки кількість оригінальних зображень зростає. Тому ми пропонуємо використовувати крок обрізки, заснований на структурі індексації, де найбільш вірогідні оригінальні зображення ефективно відбираються, а решта оригіналів відкидаються. Ми називаємо кандидатів найбільш вірогідними оригінальними зображеннями. В ідеалі, набір кандидатів містить один елемент, якщо тестове зображення дійсно є дублікатом, і жоден інший варіант. Тим не менш, більш реалістичною метою є наявність набору, розмір якого становить фіксовану частку від загальної кількості оригінальних зображень.

Algorithm 1 Multiple images duplicate detection

Require: test image $\mathbf{T}$, trained pre-classifier and set of binary detectors
Ensure: label l of most probable original, and the corresponding probability p

procedure DETERMINE_ORIGINAL($\mathbf{T}$)
  $\mathbf{f}$ = FEATURE_EXTRACTION($\mathbf{T}$) ▷ *step 1* — feature extraction
  $\mathcal{C}$ = PRE_CLASSIER($\mathbf{f}$) ▷ *step 2* — set of candidates
  for $i = 1$ to $|\mathcal{C}|$ **do**
    $m = \mathcal{C}_i$
    p_i = BINARY_DETECTOR$_m$($\mathbf{f}$) ▷ *step 3* — binary detectors
  if $|\mathcal{C}| > 0$ **then**
    $m = \arg\max_i p_i$ ▷ *step 4* — most probable original
    $l = \mathcal{C}_m$
    $p = p_m$
    return (l, p)
  else
    return $(-1, 1)$

3.2.2 Огляд системи

Тепер ми наводимо суть запропонованої системи виявлення дублікатів кількох оригінальних зображень. Система складається з чотирьох кроків, кожна з них наведена далі. Алгоритм 1 дає псевдокод механіки системи. Нагадаємо, що мета системи - визначити, чи є тестове зображення дублікатом одного оригінального зображення, відомого системі, кожне з яких позначено від 1 до N. Таким чином, алгоритм повертає передбачувану мітку $l \in \{-1, +1, \ldots, N\}$ та відповідну ймовірність p. Орієнтовна мітка -1 означає, що тестове зображення було відкинуто попереднім класифікатором і означає, що тестове зображення вважається не пов'язаним з жодним з оригіналів. Як уже зазначалося в частині 3.2.2, рішення можна отримати, порівнявши оцінену ймовірність p із фіксованим порогом u. Якщо ймовірність перевищує порогову величину, то передбачуваний оригінал - це той, який дається міткою, тоді як інакше тестове зображення вважається не пов'язаним з жодним із оригінальних зображень.

Попередня обробка та вилучення функцій Перший крок полягає у попередній обробці зображення, а потім у вилученні описових ознак із попередньо обробленого зображення. Операції попередньої обробки та вилучення характеристик ідентичні операціям, що використовуються для бінарних детекторів, і докладно описані в частині 4.4.1 та частині 4.4.2 відповідно.

Попередній класифікатор Другий крок, фактично головний внесок цієї глави, спрямований на ефективний відбір обмеженої кількості потенційних оригіналів серед усіх оригінальних зображень, відомих системі. Точніше, ми позначимо через C набір кандидатів. Оскільки набір оригінальних зображень задано O, C є підмножиною O. В ідеалі, C містить лише кілька елементів і включає правильний оригінал, якщо тестове зображення дійсно є дублікатом одного з оригіналів. Попередній класифікатор побудований навколо структури індексування. Точніше, оцінка підпростору, що охоплюється дублікатами, див.

Частина 4.1 для більш детальної інформації, індексується для кожного оригіналу.

Бінарні детектори На третьому етапі бінарні детектори, розроблені в главі 5, використовуються для впорядкування елементів у наборі кандидатів від найбільш вірогідного до найменш вірогідного оригіналу. Точніше, оцінюються ймовірності p_i, що тестове зображення є дублікатом оригіналів C_i. Нарешті, елементи набору кандидатів сортуються відповідно до відповідних ймовірностей.

Рішення Останній крок вибирає найбільш вірогідний оригінал, а також виводить відповідну ймовірність.

3.2.3 Зауваження щодо навчання

Етап попередньої класифікації потребує навчання, а саме передбачувані підпростори повинні бути проіндексовані. Навчання виконується самостійно на кожному оригіналі. Це означає, що нові оригінальні зображення можна додавати без перекваліфікації оригінальних зображень, вже проіндексованих у попередньому класифікаторі.

Процедура навчання вимагає лише позитивних прикладів для індексації оцінки підпростору, що охоплюється дублікатами. Однак для оцінки отриманої індексації потрібні як позитивні, так і негативні приклади. Для цього використовуються ті самі приклади навчання, що і для бінарного детектора. Точніше, 200 позитивних прикладів наведено в таблиці 5.3, тоді як 500 негативних прикладів отримано шляхом випадкового відбору прикладів із колекції зображень. Для отримання додаткової інформації читач повертається до частинів 5.2 та 5.5.1.

3.2.4 Попередній класифікатор

У цій частині ми детально описуємо запропонований попередній класифікатор та відповідну процедуру навчання. Наведено псевдокод попереднього класифікатора - це алгоритм 2. Алгоритм попереднього класифікатора повертає встановлені кандидати C, пов'язані з тестовим зображенням I. Крім того, алгоритм попереднього класифікатора можна параметризувати таким чином, що він вибирає більш-менш приклади. Це досягається шляхом модифікації значення, що позначається δ. Точніше кажучи, для δ, що дорівнює нулю процедура вибирає якомога менше кандидатів, а для великих значень вибирає більше кандидатів.

Попередній класифікатор підрозділяється на три етапи, а саме вилучення ознак, проекція ознак та пошук. Крок вилучення ознак такий самий, як і для двійкових детекторів, і він вже був представлений у частині 4.4.2. Другий крок полягає у зменшенні кількості функцій, оскільки 162 - занадто багато для ефективної схеми індексації, і це представлено в частині 3.2.4.1. Нарешті, у частині 3.2.4.2 представлена схема індексації, заснована на конкретній структурі індексації, а саме на R-Trees.

Algorithm 2 Finds the potential originals of a test image

Require: Originals to be indexed in the R-trees R_{tree} with algorithm 3 or algorithm 4

1: **procedure** PRE_CLASSIFY($\mathbf{I}$, δ)
2: $\mathbf{f}$ = FEATURE_EXTRACTION ($\mathbf{I}$)
3: $\tilde{\mathbf{f}} = \mathbf{W}_d \cdot \mathbf{f}$
4: $\mathcal{C}$ = SEARCH $\left(R_{tree}, \tilde{\mathbf{f}} \pm \delta\right)$
5: **return** $\mathcal{C}$

Проекція та нормалізація ознак для індексації

Багато функцій потрібні для того, щоб мати достатньо інформації для розрізнення дублікатів та недублікатів. Тим не менше, 162 функції занадто багато для побудови ефективної структури індексації. З цієї причини розмірність вектора ознак зменшується до d за допомогою використання PCA.

Нагадаємо, що алгоритм PCA знаходить напрямки, по яких максимізується розсіювання, або дисперсія, хмари точок [67]. Побудова матриці проекції $\mathbf{W_d}$ така.

1 - $\widetilde{\boldsymbol{W}}$: Алгоритм PCA застосовується до навчального набору, що містить особливості оригінальних зображень, для більш детальної інформації читач повертається до частинау 5.3.1, і в результаті отримується проекційна матриця $\widetilde{\boldsymbol{W}}$. Тоді прогнозовані ознаки задаються $\widetilde{\boldsymbol{W}} \cdot \mathbf{f}$.

2 - $\widetilde{\boldsymbol{W}}_{\mathrm{d}}$: PCA створює матрицю проекції 162 × 162. Іншими словами, кількість розмірів проектованих об’єктів дорівнює кількості вимірюваних об’єктів. Щоб зменшити кількість розмірностей з 162 до d, рядки матриці $\widetilde{\boldsymbol{W}}$ спочатку ранжують від напрямку, що має найбільшу дисперсію, до напрямку, що має найменшу дисперсію. Потім обираються d перші рядки матриці проекції, в результаті чого виходить матриця d × 162. Це призводить до матриці проекції $\widetilde{\boldsymbol{W}}_{\mathrm{d}}$. Цей вибір мотивований наступним чином. Напрямок, для якого розкид максимальний, відповідає також напрямку, вздовж якого середня відстань між точками максимальна. Вибираючи d найбільших розсіювачів, ми опосередковано вибираємо d напрямки, які незалежно найкраще частинаяють точки.

3 - $\mathbf{W_d}$: нарешті, прогнозовані ознаки нормуються так, що дисперсія вздовж кожного напрямку проекції дорівнює одиниці. Ця нормалізація може бути безпосередньо включена в матрицю проекції шляхом відповідного масштабування кожного рядка.

Індексація R-дерева

Обрана структура індексації базується на R-деревах [68], які є динамічними структурами, що використовуються для ефективного індексування просторових просторів. R-дерево - це дерево із збалансованою висотою із записами індексів у своїх листових вузлах, що містять вказівники на об'єкти даних. Спочатку R-дерева створювались для індексації просторових об'єктів за допомогою їх обмежувальних рам. Отже, структура R-дерева

побудована таким чином, щоб ефективно відповідати на запит на основі точки "Повернути всі записи, обмежувальні поля яких містять точку пошуку **p**", і запит на основі запиту "Повернути всі записи, обмежувальні поля яких перетинають поле пошуку **b**. "

Вибір створення попереднього класифікатора навколо індексної структури, що працює на обмежувальних полях, мотивований наступним чином. По-перше, обмежувальні коробки можна ефективно визначити. По-друге, обмежувальна коробка дуже гнучка, оскільки кожну її сторону можна самостійно регулювати. Нарешті, наведені далі алгоритми індексації можуть бути легко адаптовані до більш складних структур індексації, таких як M-Trees [69].

Схема грубої індексації

Оскільки елементи, витягнуті із зображень, демонструють певний рівень надійності проти маніпуляцій із зображеннями, ознаки дубліката локалізовані навколо ознак відповідного оригінального зображення. Отже, R-дерево, оптимізоване для виявлення дублікатів, може бути побудоване шляхом асоціювання обмежувального вікна, що охоплює всі повторювані приклади, з кожним оригінальним зображенням, відомим системі. Насправді, оскільки ми маємо справу з d-мірним простором, обмежувальні поля - це двовимірні прямокутники або узагальнені прямокутні паралелепіпеди. Вибір цих обмежувальних коробок має вирішальне значення для роботи R-дерева. Дійсно, якщо обмежувальні коробки занадто великі, багато з них перекриваються. Це призводить до великої кількості елементів у C. З іншого боку, якщо обмежувальні рамки занадто малі, дублікат може потрапити за межі обмежувальної рамки, що відповідає його оригіналу, який, таким чином, не включений у C.

Для того, щоб побудувати обмежувальну рамку, пов'язану з оригінальним зображенням, ми створюємо повторювані приклади, використовуючи набір маніпуляцій із зображеннями. Точніше, обмежувальне поле визначається двома векторами c_- і c_+, які контролюють його ступінь у кожному вимірі

$$c_{-}(\propto) = \min_{i} f_i(\propto) \tag{3.12}$$

$$c_{+}(\propto) = \max_{i} f_i(\propto) \tag{3.13}$$

Algorithm 3 Coarse indexation

Require: the original image $\mathbf{I}$, its identifier ID, the parameter $\delta \in [-1, +1]$, and the duplicate examples $\{\mathbf{D}\}_{i=1}^{D}$

Ensure: the R-tree contains the duplicate region estimation for the original

1: **procedure** COARSE_INDEXATION($\mathbf{I}$, ID, δ, $\{\mathbf{D}\}_{i=1}^{D}$)
2: **for** $i = 1$ to D **do**
3: $\mathbf{f}_i$ = FEATURE_EXTRACION ($\mathbf{D}_i$)
4: $\mathbf{c}_{-} = [\min_{i=1,\ldots,D} \mathbf{f}_i(\alpha)]_{\alpha=1}^{d}$
5: $\mathbf{c}_{+} = [\max_{i=1,\ldots,D} \mathbf{f}_i(\alpha)]_{\alpha=1}^{d}$
6: $\mathbf{s} = \mathbf{c}_{+} - \mathbf{c}_{-}$ ▷ compute the side lengths $s(\alpha)$ of the bounding box
7: INSERT ($R_{tree}, \mathbf{c}_{\pm} \pm \delta \cdot \mathbf{s}/2, ID$) ▷ algorithm INSERT in [Guttman, 1984]

де f_i (α) відповідають α-й ознаці і-го повторюваного прикладу, а c (α) позначає α-й елемент векторів c. Приклади, що використовуються для обчислення обмежувальних рам, детально описані в частині 6.3. Крім того, розмір індексованого вікна можна налаштувати, додавши δ · s / 2 до c_+ і віднявши ту саму суму від $c_–$. Тепер значення δ контролює щільність індексованого вікна навколо повторюваних прикладів. Наприклад, якщо δ більше нуля, індексоване поле більше, ніж обмежувальне поле. І навпаки, якщо δ менше нуля, індексоване поле менше, ніж обмежувальне поле. Відповідна процедура індексації наведена в алгоритмі 3.

Очікується, що вектор ознак дубліката, отриманого в результаті маніпуляцій, менш серйозних, ніж ті, що використовуються для побудови R-дерева, міститься у обмежувальному вікні, що відповідає його оригіналу. І навпаки, вектор ознак дубліката, сформованого в результаті більш серйозних маніпуляцій, зазвичай потрапляє за межі відповідного обмежувального поля. Тим не менше, його все ще можна отримати, використовуючи запит на основі вікна, використовуючи значення дельти, більше ніж нуль в алгоритмі 2. Однак це передбачає більший набір кандидатів.

Тонка індексація

Наведена раніше схема грубої індексації добре працює для маніпуляцій зі світлими зображеннями, але не вдається при більш серйозних перетвореннях [70]. З цієї причини ми зараз докладно розглядаємо більш складний метод індексації дублікатів. Основна ідея запропонованого алгоритму полягає в індексуванні поля для кожного прикладу навчання. Кожне поле частково оцінює підпростір, що охоплюється дублікатами, тоді як вся його оцінка дається їх об'єднанням.

Вищезазначені спостереження змушують нас розробити алгоритм індексації, представлений нижче. Точніше, алгоритм 4 описує конструкції підпростору, охопленого дублікатами для даного оригінального зображення. Спочатку створюються синтетичні дублікати, а елементи витягуються з них та з оригіналу. Для досягнення оцінки підпростору навколо кожного повторюваного прикладу створюється вікно. Спочатку визначаються найближчі сусіди повторюваного прикладу, використовуючи як міру вміст, обмежений кожною парою прикладів. Потім для визначення кутів вікна використовуються екстремальні координати k найближчих сусідів; параметр настройки δ дозволяє збільшити розмір вікна. Нарешті, поле індексується за допомогою процедури вставки від [71]; використовуваний ключ містить оригінальний ідентифікатор *ID*.

Далі ми використовуємо k = d, щоб зменшити ймовірність того, що приклади окремих тренувань визначають більше двох меж. Дійсно, d-мірна рамка визначається двома d-межами. Наприклад, якщо k = d - 1, є принаймні два приклади навчання, які визначають три межі, або один приклад навчання, який визначає чотири межі. Параметр σ, що керує розмірами обмежувальних вікон, можна розглядати як параметр регуляризації, див. Частина 2.3.1 для отримання додаткової інформації. Дійсно, велике значення σ відповідає великим обмежувальним вікнам і, отже, відносно грубішій оцінці дубліката частинау. З іншого боку, невелике значення σ відповідає малим обмежувальним вікнам і, отже, більш точній оцінці дубліката частинау. Це також означає, що чим менше

значення δ, тим вищий ризик перетренованості. Значення δ, отже, є досить критичним для отримання хороших показників. Отже, значення σ вибирається за допомогою процедури перехресної перевірки, подібної до тієї, що використовується в частині 5.3.3. Іншими словами, σ вибирається як той, який максимізує F-оцінку.

Algorithm 4 Fine indexation

Require: the original image $\mathbf{I}$, its identifier ID, the parameters $\delta \in [-1, +1]$ and k, and the duplicate examples $\{\mathbf{D}\}_{i=1}^{D}$

Ensure: the R-tree contains the duplicate region estimation for the original

procedure FINE_INDEXATION($\mathbf{I}$, ID, δ, k, $\{\mathbf{D}\}_{i=1}^{D}$)
  for $i = 1$ to D **do**
    $\mathbf{f}_i$ = FEATURE_EXTRACTION $(\mathbf{D}_i)$
  for $i = 1$ to D **do** ▷ add a box per duplicate example
    $\boldsymbol{\sigma}$ = ORDER_BY_CONTENT $\left(\{\mathbf{f}_j\}_{j=1}^{D}, \mathbf{f}_i\right)$
    $\mathbf{c}_- = \left[\min_{j=1,\ldots,k} \mathbf{f}_{\boldsymbol{\sigma}(j)}(\alpha)\right]_{\alpha=1}^{d}$
    $\mathbf{c}_+ = \left[\max_{j=1,\ldots,k} \mathbf{f}_{\boldsymbol{\sigma}(j)}(\alpha)\right]_{\alpha=1}^{d}$
    $\mathbf{s} = \mathbf{c}_+ - \mathbf{c}_-$ ▷ compute the side lengths $\mathbf{s}(\alpha)$ of the bounding box
    INSERT $(R_{tree}, \mathbf{c}_\pm \pm \delta \cdot \mathbf{s}/2, ID)$ ▷ algorithm INSERT in [Guttman, 1984]
procedure ORDER_BY_CONTENT($\{\mathbf{c}_i\}_{i=1}^{N}, \mathbf{c}$)
  for $i = 1$ to N **do**
    $v_i = (\mathbf{c}_i - \mathbf{f})^T \cdot (\mathbf{c}_i - \mathbf{f})$
  $\boldsymbol{\sigma}$ = SORT$(\{v_i\}_{i=1}^{N})$ ▷ $\boldsymbol{\sigma}$ is a permutation of $1, \ldots, N$ s.t. $v_{\boldsymbol{\sigma}(i)} \geq v_{\boldsymbol{\sigma}(i-1)}$
  return $\boldsymbol{\sigma}$

3.2.5 Результати для попереднього класифікатора

У цій частині ми представляємо експериментальні результати для оцінки ефективності запропонованого попереднього класифікатора. Перший експеримент, представлений у частині 3.2.5.1, порівнює запропонований попередній класифікатор із системою, заснованою на стандартній метриці L_1. Другий експеримент, описаний у частині 3.2.5.2, досліджує масштабованість запропонованого попереднього класифікатора.

Масштабованість

Тепер ми звернемо нашу увагу на масштабованість запропонованого попереднього класифікатора. Під масштабованістю ми розуміємо поведінку розміру набору кандидатів як кількості оригінальних зображень, відомих

вирощеній системі. В ідеалі попередній класифікатор підтримує постійну кількість кандидатів, оскільки кількість оригінальних зображень, відомих системі, зростає. Наприклад, це випадок із методом k найближчого сусіда на основі L_1. На жаль, йому потрібно досить велике значення k, щоб досягти низьких коефіцієнтів пропусків для складних перетворень. Більш реально, хороший попередній класифікатор зводить кількість кандидатів до частки від загальної кількості зображень, відомих системі.

Масштабованість - експериментальна установка

Щоб перевірити масштабованість системи, ми використовуємо ті самі колекції зображень, а також ті самі тести, що і в частині 3.2.5.1. Аналогічно, N оригінальних зображень індексується за допомогою алгоритму 4 з d = 30, як описано в частині 6.4. У цьому експерименті кількість оригінальних зображень N спочатку встановлюється на 25, потім на 50, 100 та 200. У випадку N = 200 двісті зображень збігаються з тими, що використовувались у частині 6.5.1. З іншого боку, оригінальні зображення для інших випадків (N <200) випадковим чином вибираються серед цих двохсот зображень. Потім у кожному випадку частоту промахів і частоту потраплянь вимірюють для різного розміру вікна пошуку, див. Частина 6.5.1 для отримання додаткової інформації. Для отримання плавних кривих експерименти проводяться п’ять разів у випадках, коли N <200. Для кожного циклу відбираються різні вихідні зображення; і результати остаточно усереднюються.

Масштабованість - MM270k та CGFA

На рисунку 6.3 показані результати масштабованості для колекції зображень MM270k. Перший рядок (рисунок 6.3a та рисунок 6.3b) надає частоту потрапляння, середній розмір набору кандидатів C, в залежності від частоти пропусків, тоді як другий рядок (малюнок 6.3c та малюнок 6.3d) подає ту саму інформацію, але нормалізовану відносно кількості оригінальних зображень, відомих системі. Подібним чином у першому стовпці (рисунок 6.3a та малюнок 6.3c) подається частота потрапляння, знайдена для повторюваних тестових зображень, тоді як у другому стовпці (малюнок 6.3b та рисунок 6.3d)

відображається частота потрапляння, отримана для не пов'язаних тестових зображень. Перший стовпець додатково відображає ту саму інформацію, отриману за допомогою попереднього класифікатора k-найближчого сусіда (KNN) на основі L_1.

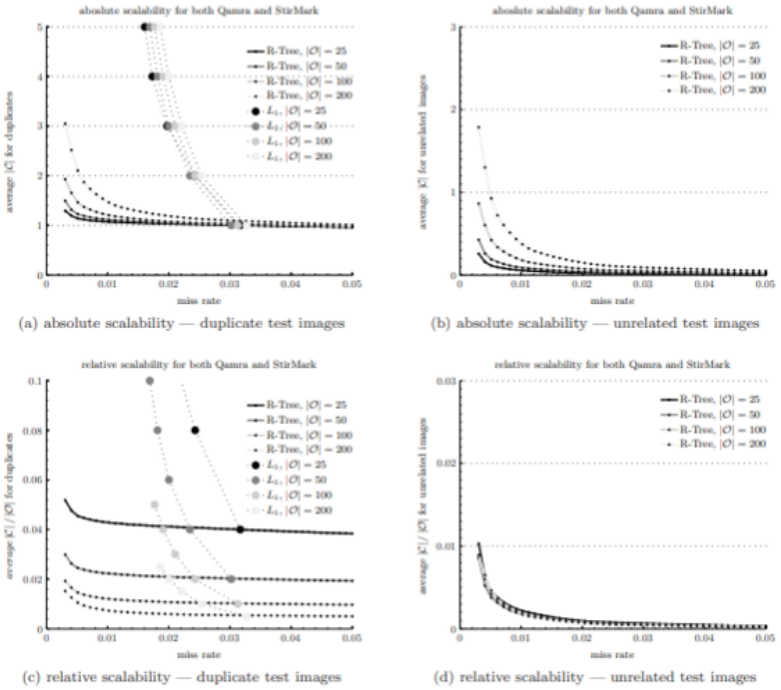
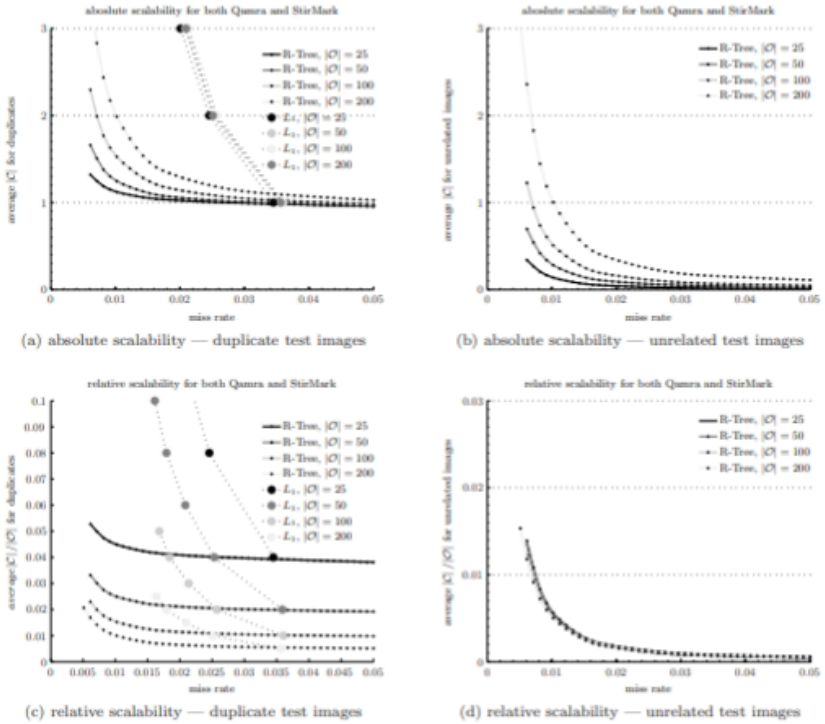


Рисунок 3.7 Колекція MM270k - масштабованість попереднього класифікатора. Цей малюнок показує масштабованість, отриману за допомогою попереднього класифікатора на основі L_1, порівняно з тим, який досягається запропонованим попереднім класифікатором. Використовуються як тести Qamra, так і StirMark.

Масштабованість - висновок

Запропонований попередній класифікатор масштабується, а також збільшується кількість оригінальних зображень. Дійсно, він повертає в середньому фіксовану частку від загальної кількості оригінальних зображень незалежно від кількості оригінальних зображень, відомих системі. Наприклад, для середньої частоти пропусків 0,005 попередній класифікатор повертає 0,005

· N потенційних кандидатів в середньому, де N - загальна кількість оригінальних зображень. Іншими словами, 99,5 відсотків оригінальних зображень відкидаються.



(a) absolute scalability — duplicate test images (b) absolute scalability — unrelated test images

(c) relative scalability — duplicate test images (d) relative scalability — unrelated test images

Рисунок 3.8 Коллекція CGFA - масштабованість попереднього класифікатора. Цей малюнок показує масштабованість, отриману за допомогою попереднього класифікатора на основі L_1, порівняно з тим, який досягається запропонованим попереднім класифікатором. Використовуються як тести Qamra, так і StirMark.

3.2.6 Результати для системи

У цій частині ми представляємо експериментальні результати для оцінки запропонованого дублюючого детектора. Перший експеримент, представлений у частині 6.6.1, представляє ефективність, отриману повним детектором дублікатів кількох оригінальних зображень. Другий експеримент, що

розглядається в частині 6.6.2, представляє простір для зберігання та обчислювальний ресурс, необхідний запропонованій системі виявлення дублікатів. Останні експерименти, описані в частині 6.6.3, аналізують ефективність запропонованої системи щодо двох інших сучасних методів.

Порівняння з існуючими методами виявлення дублікатів

Тепер ми порівняємо ефективність запропонованого методу з ефективністю існуючих систем виявлення дублікатів. Використовуються ті самі існуючі роботи, що і в главі 3, а саме ключові моменти (КП) [72] та функція сприймання відстані (DPF) [73].

Порівняння - результати та аналіз

Рисунок 3.2.7 порівнює ефективність запропонованої системи виявлення дублікатів із сучасними технологіями, про які повідомляється в [2;74]. Чорна лінія відповідає кривій DET, отриманій у нашій системі. Світло-сіра лінія відображає ефективність методу виявлення дублікатів на основі DPF [75]. Хрестик вказує на ефективність системи виявлення дублікатів на основі KP [76].

Тепер, порівнюючи ефективність методу DPF з ефективністю запропонованої системи, можна спостерігати дві речі. По-перше, метод DPF не досягає помилки FN для хибнопозитивних показників помилок вище 1×10^{-3}. Однак, опинившись нижче цієї точки, продуктивність надзвичайно швидко погіршується. По-друге, хоча DPF працює дещо краще, ніж запропонована система для випадків помилково позитивних помилок вище 1×10^{-3}, вона явно перевищує цей поріг.

З іншого боку, запропонований метод перевершує КП. Дійсно, для колекції CGFA, KP досягає коефіцієнта помилок FNs $1{,}5 \times 10^{-3}$ без жодної помилково позитивної помилки. З іншого боку, запропонований метод ніколи не досягає, для того самого набору тестів, жодної помилково позитивної помилки. Однак розрив у продуктивності є досить низьким для колекції MM270k, але ці результати менш значні, оскільки вони екстраполюються для

методу KPs, і, можливо, його ефективність завищена. Пояснення для кращих результатів КП такі самі, як і в частині 4.

3.2.7 Випадкова проекція

У частині помічено, що попередній класифікатор працює ефективніше, коли для індексації дубліката використовується мала кількість вимірів, а саме тридцять, ніж коли використовується більше вимірів, а саме шістдесят. Це хороший приклад ефекту прокляття розмірності [77]. З іншого боку, попередній класифікатор на основі L_1 поводиться подібним чином, але оптимальна кількість розмірів у цьому випадку вища, а саме вісімдесят. Отже, різниця в п’ятдесят в оптимальній кількості розмірів показує, що додані функції, у випадку L_1, все ще несуть дискримінаційну інформацію. Тоді виникає питання, як ефективно використовувати цю додаткову інформацію?

Індексація багатовимірного простору широко вивчалася, і існує безліч робіт щодо того, як уникнути або зменшити прокляття розмірності. Тим не менше, не існує жодного методу, який би це повністю вирішив. Популярне рішення, яке працює досить добре, називається локально чутливим хешем [78;79]. Коротше кажучи, цей підхід полягає у випадковому проектуванні об’єктів на менший простір шляхом випадкового вибору підмножини всього набору об’єктів. Таким чином, єдиний вектор ознак представлений безліччю випадкових проекцій. Нарешті, кожну проекцію можна проіндексувати в межах невеликого простору розмірності і, отже, уникнути прокляття розмірності. Пізніше результат запиту до бази даних складається із записів, кілька випадкових проекцій яких відповідають запитам.

Цей підхід можна застосувати до попереднього класифікатора, представленого в цій главі, оскільки він базується на індексації. Однак адаптація алгоритму та експерименти необхідні для того, щоб визначити вплив випадкової проекції на ефективність попереднього класифікатора.



Рисунок 3.9 Виправлення зображень. Цей малюнок представляє ділянки зображення на різних рівнях деталізації.

Ієрархічне виявлення дублікатів

Ми пропонуємо аналізувати зображення на різних рівнях деталізації l. На кожному рівні деталізації зображення поділяється на ділянки однакового розміру. Наприклад, на найгрубішому рівні деталізації є один патч розміру зображення, на наступному рівні - 4 патчі, потім 9 патчів тощо. На рисунку 3.2.8 показані ці патчі для трьох перших рівнів деталізації.

Далі ми пояснимо потенціал різних рівнів деталізації, які використовуються для опису зображення. Очевидно, що зображення складається з різних областей, кожна з яких має різні характеристики, як видно на малюнку 6.8. Глобальні особливості, особливості, що описують зображення в цілому, дають усереднену версію характеристик кожного регіону і добре працюють для виявлення дублікатів [81;80]. Однак можливо, що не пов’язані між собою зображення мають дуже схожі глобальні риси, і в цьому випадку вони будуть вважатися дублікатами один одного. Використання описів зображень із рівнями деталізації дозволяє зменшити кількість таких зіткнень. Незважаючи на те, що непов’язані зображення можуть мати подібні глобальні функції, менш імовірно, що більшість їхніх патчів мають подібні функції. Це основна ідея, яка лежить в основі ієрархічного підходу, запропонованого нижче. Підпростір, охоплений дублікатами, будується для кожного рівня деталізації. Потім потенційні оригінали тестового зображення визначаються, використовуючи лише глобальний рівень деталізації. Кожні патчі тестового

зображення згодом перевіряються на більш тонкому частині деталізації кожного оригіналу, знайденого раніше, і відповідний оригінал зберігається лише в тому випадку, якщо кількість відповідних патчів достатня. Нарешті, операцію можна повторити для решти оригіналів до досягнення найтоншого рівня деталізації, ми експериментували до L = 3. Наведені вище спостереження змусили нас розробити ієрархічний попередній класифікатор, представлений далі.

Ієрархічний попередній класифікатор

Запропоноване розширення попереднього класифікатора працює в ієрархічному порядку. Він починається з рівня l = 1 і продовжується на більш точних рівнях деталізації, можливо до l = L. Початковий набір кандидатів визначається на рівні l = 1. На більш точних рівнях деталізації цей набір обрізається шляхом видалення оригіналів із недостатньою кількістю відповідних патчів. Операцію повторюють до досягнення найдрібнішого рівня деталізації. Алгоритм 5 дає псевдокод ієрархічного класифікатора. Він використовує попередній класифікатор, заданий алгоритмом 2.

Алгоритм 6 описує конструкції безлічі повторюваних різновидів для даного оригінального зображення. На кожному рівні деталізації l приклади навчання поділяються на $(l + 1)^2$ патчі. Потім кожен патч описується вектором ознак. Потім дублюючий колектор оцінюється для кожного патча на заданому рівні деталізації. Для цього використовується алгоритм 4.

Algorithm 5 Finds the potential originals of a test image

Require: originals to be indexed in the R-tree with algorithm 6

procedure HIERARCHICAL_PRE_CLASSIFIER($\mathbf{I}$, $\boldsymbol{\delta}$, $\mathbf{m}$)
  for $l = 1$ to L **do**
    $\mathcal{R} = \emptyset$
    for $b = 1$ to $(l+1)^2$ **do** ▷ treat each patch separately
      $\tilde{\mathbf{I}} = \text{GET_PATCHES}(\{\mathbf{D}\}_{i=1}^{D}, l, b)$
      $\{(ID_i, \tilde{l}_i, \tilde{b}_i)\}_i = \text{PRE_CLASSIFY}\left(\tilde{\mathbf{I}}, \boldsymbol{\delta}_l\right)$ ▷ see algorithm 2
      $\mathcal{R} = \mathcal{R} \bigcup \{(ID_i, \tilde{l}_i, \tilde{b}_i)\}_i$
    if $l = 1$ **then**
      $\mathcal{C} = \{ID \text{ such that } (ID, l, 1) \in \mathcal{R}\}$ ▷ determine the initial set potential originals
    else
      for $ID \in \mathcal{C}$ **do**
        if $|\{b \text{ such that } (ID, l, b) \in \mathcal{R}\}| < \mathbf{m}(l)$ **then**
          $\mathcal{C} = \mathcal{C} \setminus ID$ ▷ not enough patches match
  return $\mathcal{C}$

Algorithm 6 Estimates a set of duplicate manifolds of an original and indexes it

Require: the original image $\mathbf{I}$, its identifier ID, the parameters $\delta \in [-1, +1]$, k and L, and the duplicate examples $\{\mathbf{D}\}_{i=1}^{D}$

Ensure: the R-tree contains the duplicate region estimation for the original

procedure HIERARCHICAL_INDEXATION($\mathbf{I}$, ID, δ, k, L, $\{\mathbf{D}\}_{i=1}^{D}$)
  for $l = 1$ to L **do** ▷ treat each level separately
    for $b = 1$ to $(l+1)^2$ **do** ▷ treat each patch separately
      $\{\tilde{\mathbf{D}}\}_{i=1}^{D} = \text{GET_PATCHES}(\{\mathbf{D}\}_{i=1}^{D}, l, b)$
      FINE_INDEXATION $\left(\mathbf{I}, (ID, l, b), \delta, k, \{\mathbf{D}\}_{i=1}^{D}\right)$ ▷ see algorithm 4

Результати та зауваження

Запропонований ієрархічний алгоритм реалізований і був представлений з більш детальною інформацією в [82]. Попередні результати обнадійливі, оскільки середній розмір набору кандидатів для непов'язаних тестових зображень може бути зменшений у 2,6 рази для двох рівнів деталізації та в 3,4 разів для трьох рівнів деталізації. Більше того, слід зазначити, що цей ієрархічний підхід також може застосовуватися до бінарних детекторів. Звичайно, застосовність такої ієрархічної системи вимагає подальших досліджень.

3.2.8 Підсумок глави

У цій главі ми представили систему виявлення дублікатів декількох оригінальних зображень на основі двійкового детектора, представленого раніше в главі 3.2. Для створення ефективної системи тестове зображення перевіряється лише на двійкових детекторах, що відповідають найбільш

вірогідним оригінальним зображенням. Їх вибір із використанням попереднього класифікатора є головним внеском цієї глави. Потім аналізується ефективність попереднього класифікатора. Згодом вся система аналізується та порівнюється з найсучаснішими методами. Нарешті, пропонується можливе вдосконалення системи.

Пропонована система складається з п'яти кроків, викладених далі. На першому кроці для опису зображення використовується глобальна статистика. На другому кроці кількість функцій зменшується. На третьому етапі найбільш вірогідні оригінали відбираються за допомогою R-дерева. Вони формують набір кандидатів. На чотирьох кроках двійкові детектори, розроблені в главі 5, застосовуються до кожного елементу набору кандидатів. Нарешті, вибирається елемент з найбільшою ймовірністю, і система оцінює тестове зображення як копію відповідного оригіналу. Система також надає оцінку ймовірності правильності цього вибору.

Ефективність запропонованої системи оцінюється за допомогою стандартних еталонів та аналізується результат. З’ясувалося, що запропонований детектор, що виявляє дублікати оригінальних зображень, значно перевершує детектори, що використовують ті самі функції, але на основі метрики L_1. Система додатково порівнюється з найсучаснішими методами виявлення дублікатів. Він перевершує метод DPF, який використовує більше можливостей для опису зображення. Хоча запропонований метод дещо перевершує метод KP, він обчислювально менш складний у п’ять-десять разів.

Нарешті, продуктивність запропонованої системи можна значно покращити, використовуючи ієрархічну систему, яка поділяє зображення на піраміду патчів і створює детектори, налаштовані на кожен патч. Другий напрямок досліджень стосується попереднього класифікатора, який має кращу ефективність у просторових просторах. Однак ці напрямки досліджень вимагають подальших робіт.

ВИСНОВКИ

У частині 1 ми побачили, що проблема виявлення дублікатів зображень походить з різних полів, а саме водяних знаків та пошуку на основі вмісту. Потім були розглянуті плюси і мінуси кожного підходу. В основному виявлення дублікатів на основі вмісту є більш гнучким, але ще не таким зрілим, як водяні знаки, з точки зору точності та швидкості повнотау. Крім того, властивою слабкістю будь-якої техніки, що базується на вмісті, є неможливість розрізнити фотографії тієї самої сцени, зроблені з дещо інших ракурсів або в різний час. З іншого боку, водяні знаки менш гнучкі, ніж метод, що базується на вмісті, оскільки вимагає модифікації зображення до його розповсюдження. Ця вимога є причиною двох основних недоліків водяних знаків, як описано далі. По-перше, водяні знаки адаптуються лише в тому випадку, якщо людина має повний контроль над оригінальною ілюстрацією та готова її змінити. По-друге, зображення з водяним знаком можна виявити, доки не буде виявлено засіб для ефективного видалення водяного знаку. Після того, як водяний знак буде видалено із зображення, визначити копії цього немаркованого зображення за допомогою методів водяних знаків однозначно неможливо. Загалом, підходи, пов’язані з водяними знаками та вмістом, досить доповнюють.

У частині 2 ми розробили основу для систем виявлення дублікатів на основі вмісту. Рамка виявлення дублікатів спочатку складається з моделі підпростору, охопленої дублікатами оригінального зображення. Ця модель дозволяє дослідити деякі характеристики дублікатів зображення; наприклад, встановлено, що за певних припущень дублікати утворюють різноманіття, вбудоване в простір зображення. Другим елементом фреймворку є загальна система виявлення дублікатів. Завдяки цій загальній системі ми розробляємо наш погляд на виявлення дублікатів, а саме класифікацію тестового зображення в один із класів K + 1. K-класи відповідають K-оригінальним зображенням, відомим системі, або іншими словами "тестове зображення є дублікатом одного з відомих оригіналів", тоді як решта класів означає "тестове зображення не

пов'язане ні з одним із відомих оригіналів зображення ". Нарешті, останній елемент системи стосується методології оцінки системи виявлення дублікатів, що базується на представленому підході до класифікації.

Ще в частині 2 ми дали огляд фактичної системи виявлення дублікатів, розробленої в цій дипломній роботі. Система складається з чотирьох етапів, а саме вилучення особливостей, попередній класифікатор, двійкові детектори-повторювачі та остаточне рішення. Вилучення ознак полягає в описі зображень за допомогою відповідної візуальної статистики. Попередній класифікатор націлений на вибір обмеженої кількості оригіналів серед K-оригінальних зображень; оригінал вибирається, якщо тестове зображення потенційно є одним із його дублікатів. Двійкові детектори дублікатів складаються з двійкових класифікаторів, що використовуються для визначення ймовірності того, що тестове зображення є дублікатом кожного вибраного оригінального зображення. На останньому етапі рішення просто полягає у виборі найбільш вірогідного оригіналу.

У частині 3 ми представили наш двійковий детектор дублікатів. Основна ідея запропонованого детектора полягає у адаптації виявлення дублікатів до конкретного оригінального зображення. Потім система може класифікувати тестові зображення як дублікати оригінального зображення або як не пов’язані між собою зображення. Бінарний детектор використовує опис зображення, наведений у частині 2, і складається з трьох етапів, описаних далі. На першому кроці елементи проектуються лінійно, щоб отримати кращий розділ між дублікатами оригінального зображення та не пов’язаних між собою зображень. На другому етапі елементи проектованої ознаки нормуються відповідно до статистичного розподілу дублікатів. На останньому кроці для визначення ймовірності того, що тестове зображення є дублікатом оригінального зображення, використовується нелінійна функція прийняття рішення на основі класифікатора опорних векторів. Оцінюється ефективність запропонованої системи та аналізуються результати. З'ясовано, що запропонований дублюючий детектор на основі SVC значно перевершує детектори, що використовують ті

самі функції, але на основі метрики L_1. Потім пропонований двійковий детектор порівнюють із сучасною системою. Він перевершує метод функції сприймання відстані (DPF), який використовує більше можливостей для опису зображення. Хоча запропонований метод дещо перевершує метод ключових точок (КП), він обчислювально простіший в п'ять-десять разів.

У частині 3.2 ми розповіли про всю систему дублікатів. На відміну від частини 3, система знає набір оригінальних зображень, а не лише один оригінал. Запропонована система використовує опис зображення, наведений у главі 2, і додатково складається з трьох етапів, викладених далі. На першому кроці кількість функцій зменшується. На другому етапі оригінали, які, швидше за все, будуть дублікатами тестового зображення, відбираються за допомогою R-Tree. Вони формують набір кандидатів. На третьому етапі бінарні детектори, розроблені в главі 2, застосовуються до кожного елементу набору кандидатів. Нарешті, вибрано елемент з найбільшою ймовірністю. Система оцінює, що тестове зображення є дублікатом цього оригіналу, якщо відповідна ймовірність перевищує певний поріг. Оцінюється ефективність запропонованої системи та аналізуються результати. З’ясувалося, що запропонований детектор, що виявляє дублікати оригінальних зображень, значно перевершує детектори, що використовують ті самі функції, але на основі метрики L_1. Крім того, він також перевершує метод DPF, який використовує більше функцій для опису зображення. Хоча запропонований метод дещо перевершується методом KP, він обчислювально менш складний у п’ять-десять разів.

На завершення цього висновку я хотіла б зазначити, що природа цього дипломув основному є дослідницькою. Дійсно, наскільки мені відомо, це одна з перших спроб застосувати методи машинного навчання до проблеми виявлення дублікатів на основі вмісту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] E. L. Allwein, R. E. Schapire, Y. Singer (2000). Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. Journal of Machine Learning Research 1:113–141.

[2] M. Barnett (1999). Digital watermarking: applications, techniques and challenges. IEEE Electronics & Communication Engineering Journal 11(4):173–183.

[3] M. Barni (2003a). What is the future for watermarking? (part I). Signal Processing Magazine, IEEE 20(5):55–60.

[4] M. Barni (2003b). What is the future for watermarking? (Part II). Signal Processing Magazine, IEEE 20(6):53–59.

[5] R. Bayer, E. M. McCreight (1972). Organization and maintenance of large ordered indexes. Acta Informatica V1(3):173–189.

[6] N. Beckmann, H.-P. Kriegel, R. Schneider, B. Seeger (1990). The R*-tree: An efficient and robust access method for points and rectangles. In ACM SIGMOD International Conference on Management of Data, pp. 322–331.

[7] J. L. Bentley (1975). Multidimensional Binary Search Trees Used for Associative Searching. Commun. ACM 18(9):509–517. C. Boehm, S. Berchtold, D. Keim (2001). Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Comput. Surv. 33(3):322–373.

[8] R. Boynton (1989). Eleven colors that are almost never confused. In Proceedings of the SPIE Symposium Human Vision, Visual Processing, and Digital Display, vol. Volume 1077, pp. 322– 332, Bellingham.

[9] L. Breiman (1996). Bagging Predictors. Machine Learning 24(2):123–140. C. J. C. Burges (1998). A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2(2):121–167.

[10] C. Chang, C. Lin (2007). LIBSVM: a Library for Support Vector Machines.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16:321—357.

[12] P.-H. Chen, C.-J. Lin, B. Sch¨olkopf (2005). A tutorial on nu-support vector machines. Applied Stochastic Models in Business and Industry 21:111–136.

[13] P. Ciaccia, M. Patella, P. Zezula (1997). M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In VLDB ’97: Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 426–435, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[14] Cox, M. Miller, J. Bloom (2001). Digital Watermarking: Principles & Practice. Morgan Kaufmann.

[15] J. Cox, M. L. Miller (2002). The First 50 Years of ElectronicWatermarking. EURASIP Journal on Applied Signal Processing 2:126–132.

[16] F. C. Crow (1984). Summed-Area Table for Texture Mapping. Computer Graphics 18(3):207–212.

[17] S. R. Deans (1983). The Radon Transform and Some of Its Applications. Krieger Publishing Company.

[18] Y. Deng et al. (2001). An efficient color representation for image retrieval. Image Processing, IEEE Transactions on 10(1):140–147.

[19] D. L. Donoho (1998). High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality.

[20] K. Duan, S. S. Keerthi, A. N. Poo (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing 51:41–59.

[21] R. Duda, P. Hart, D. Stork (2001). Pattern Classification. Wiley-Interscience.

[22] D. J. Elzinga, D. W. Hearn (1972). The minimum covering sphere problem. Management Science 19:96–104.

[23] R. Fagin, J. Nievergelt, N. Pippenger, H. R. Strong (1979). Extendible hashing — a fast access method for dynamic files. ACM Trans. Database Syst. 4(3):315–344.

[24] C. Faloutsos et al. (1994). Efficient and Effective Querying by Image Content. Journal of Intelligent Information Systems 3(3/4):231–262.

[25] T. Fawcett (2003). ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. In Technical Report HPL-2003-4, HP Labs.

[26] G. Finlayson, G. Schaefer (2001). Hue that is Invariant to Brightness and Gamma. In Proc. British Machine Vision Conf., pp. 303–312, Manchester, England.

[27] M. M. Fleck, D. A. Forsyth, C. Bregler (1996). Finding Naked People. In ECCV(2), pp. 593–602.

[28] J. Fridrich (1999). Robust Bit Extraction from Images. In IEEE International Conference On Multimedia Computing And Systems, IEEE, Florence.

[29] J. Fridrich (2000). Visual Hash for Oblivious Watermarking. In SPIE Photonic West Electronic Imaging, Security and Watermarking of Multimedia Contents Conference, San-Jose.

[30] V. Gaede, O. Guenther (1998). Multidimensional access methods. ACM Comput. Surv. 30(2):170– 231.

[31] R. Gonzalez, R. Woods (2002). Digital Image Processing, 2/E. Prentice-Hall.

C. Gotlieb, H. Kreyszig (1994). Texture descriptors based on co-occurence matrices. Computer vision, graphics and image processing 51:70–86.

[32] M. Grabner, H. Grabner, H. Bischof (2006). Fast approximated SIFT. In Lecture Notes in Computer Science, vol. 3851, pp. 918–927, Springer.

[33] R. Haralick, K. Shanmugam, I. Dinstein (1973). Texture features for image classification. IEEE Transaction on System, Man, and Cybernetic 3(6):610–621.

[34] F. Hartung, M. Kutter (1999). Multimedia watermarking techniques. Proceedings of the IEEE 87(7):1079 – 1107.

[35] C.-Y. Hsu, C.-S. Lu (2004). Geometric distortion-resilient image hashing system and its application scalability. In Proceedings of the 2004 workshop on Multimedia and security, pp. 81–92, New York.

[36] M.-K. Hu (1962). Visual Pattern Recognition by Moment Invariants. IEEE Transaction on Information Theory 8:179–187.

[37] P. Indyk, R. Motwani (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In Proceedings of the ACM symposium on Theory of computing, pp. 604–613, ACM Press, New York, NY, USA.

[38] J. Jackson (1991). A User's Guide to Principal Components.

[39] M. Johnson, K. Ramchandran (2003). Dither-based secure image hashing using distributed coding. In Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, vol. 2, pp. II–751–4 vol.3.

[40] T. Kalker, J. Haitsma, J. Oostveen (2001). Issues with digital watermarking and perceptual hashing. In Proceedings of SPIE - The International Society for Optical Engineering, vol. 4518, pp. 189–197, Philips Research Eindhoven, Prof. Holstlaan 4, 5656 AA, Eindhoven, Netherlands.

[41] J. Karhunen, J. Joutsensalo (1994). Representation and separation of signals using nonlinear PCA type learning. Neural Netw. 7(1):113–127.

[42] Y. Ke, R. Sukthankar (2004). PCA-SIFT: a more distinctive representation for local image descriptors. In Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, vol. 2, pp. II–506–II–513Vol.2.

[43] Y. Ke, R. Sukthankar, L. Huston (2004). Efficient Nearduplicate Detection and Subimage Retrieval. In ACM International Conference on Multimedia, pp. 869–876.

[44] S. S. Keerthi, C.-J. Lin (2003). Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Computation 15(7):1667–1689.

[45] C. Kim (2003). Content-based image copy detection. Signal Processing: Image Communication 18(3):169–184.

[46] H.-P. Kriegel (1984). Performance comparison of index structures for multi-key retrieval. In SIGMOD '84: Proceedings of the 1984 ACM SIGMOD international conference on Management of data, pp. 186–196, ACM Press, New York, NY, USA.

[47] N. Kwak, c.-H. Choi (2003). Feature extraction based on ICA for binary classification problems. IEEE Transactions on Knowledge and Data Engineering 15(6):1374–1388.

[48] D. T. Lee, B. J. Schachter (1980). Two algorithms for constructing a Delaunay triangulation. International Journal of Parallel Programming V9(3):219–242.

[49] H. Lejsek, F. H. Asmundsson, B. T. J´onsson, L. Amsaleg (2006a). Scalability of local image descriptors: a comparative study. In MULTIMEDIA ’06: Proceedings of the 14th annual ACM international conference on Multimedia, pp. 589–598, ACM Press, New York, NY, USA.

[50] H. Lejsek, F. H. Asmundsson, B. Th´or-J´onsson, L. Amsaleg (2006b). Blazingly Fast Image Copyright Enforcement. In ACM International Conference on Multimedia, demonstration, pp. 489–490, Santa Barbara, CA, USA.

[51] J.-L. Leu (1991). Computing a Shape’s Moments from its Boundary. Pattern Recognition 24(10):949–957.

[52] B. Li, E. Chang, C.-T. Wu (2002). DPF - a perceptual distance function for image retrieval. In Image Processing. 2002. Proceedings. 2002 International Conference on, vol. 2, pp. II–597–II–600 vol.2.

[53] T. Lindeberg, J.-O. Eklundh (1992). Scale-space primal sketch: construction and experiments. Image Vision Comput. 10(1):3–18.

[54] D. Lowe (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2):91–110.

[55] C.-S. Lu, C.-Y. Hsu (2005). Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication. Multimedia Systems V11(2):159–173.

[56] C.-S. Lu, C.-Y. Hsu, S.-W. Sun, P.-C. Chang (2004). Robust mesh-based hashing for copy detection and tracing of images. In Multimedia and Expo 2004, vol. 1, pp. 731–734 Vol.1.

[57] B. Manjunath, W. Ma (1996). Texture Features for Browsing and Retrieval of Image Data. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(8):837–842.

[58] B. Manjunath, J.-R. Ohm, V. Vasudevan, A. Yamada (2001). Color and texture descriptors. Circuits and Systems for Video Technology, IEEE Transactions on 11(6):703–715.

[59] Y. Maret, F. Dufaux, T. Ebrahimi (2005a). Image Replica Detection based on Support Vector Classifier. In Proc. SPIE Applications of Digital Image Processing XXVIII, Santa Barbara, USA.

[60] Y. Maret, F. Dufaux, T. Ebrahimi (2006a). Adaptive Image Replica Detection based on Support Vector Classifiers. Signal Processing : Image Communication 21(8):688–703.

[61] Y. Maret, G. Molina, T. Ebrahimi (2005b). Identification of Image Variations based on Equivalence Classes. In Proc. SPIE Visual Communications and Image Processing, SPIE, Beijing, China.

[62] Y. Maret, G. G. Molina, T. Ebrahimi (2005c). Images Identification Based on Equivalence Classes. In Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005).

[63] Y. Maret, D. M. Sanjuan, F. Dufaux, T. Ebrahimi (2006b). Hierarchical Indexing using R-trees for Replica Detection. In SPIE, SPIE, SPIE.

[64] Y. Maret et al. (2006c). A Novel Replica Detection System using Binary Classifiers, R-trees, and PCA. In International Conference on Image Processing, IEEE. A. Martin et al. (1997). The DET Curve in Assessment of Detection Task Performance. In Proc. Eurospeech ’97, pp. 1895–1898, Rhodes, Greece.

[65] D. Medin, R. Goldstone, D. Gentner (1993). Respects for Similarity. Psychological Review 100(2):254–278.

[66] M. Mihcak, R. Venkatesan (2001). New Iterative Geometric Methods for Robust Perceptual Image Hashing. In ACM Workshop on Security and Privacy in Digital Rights Management, Philadelphia.

[67] V. Monga, A. Banerjee, B. Evans (2004). Clustering Algorithm for Perceptual Image Hashing. In IEEE International Conference on Image Processing, Singapore.
[68] V. Monga, A. Banerjee, B. Evans (2006). A clustering based approach to perceptual image hashing. Information Forensics and Security, IEEE Transactions on 1(1):68–79.
[69] V. Monga, B. L. Evans (2004). Robust Perceptual Image Hashing Using Feature Points. In IEEE International Conference on Image Processing, Singapore.
[70] P. Moulin (2003). Comments on “Why watermarking is nonsense”. Signal Processing Magazine, IEEE 20(6):57–59.
[71] MPEG12816 (2006). Local region descriptor robust invariant geometrical transformation.
[72] MPEG12841 (2006). Proposed test conditions for MPEG-7 Visual Core Experiments 6. MPEG13152 (2006). Visual identifier which is robust and invariant to image modification.
[73] MPEG13579 (2006). The non-geometric model of local region descriptor as the visual identifier.
[74] MPEG13861 (2006). A Test Image Management System for MPEG-7 core experiments.
[75] K. Muller et al. (2001). An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12(2):181–201.
[76] W. Niblack et al. (1993). QBIC project: querying images by content, using color, texture, and shape. In W. Niblack (ed.), Proc. SPIE Vol. 1908, p. 173-187, Storage and Retrieval for Image and Video Databases, Wayne Niblack; Ed., pp. 173–187.
[77] J. Nievergelt, H. Hinterberger, K. C. Sevcik (1984). The Grid File: An Adaptable, Symmetric Multikey File Structure. ACM Trans. Database Syst. 9(1):38–71.
[78] L. Penna, A. Clark, G. Mohay (2005). Challenges of automating the detection of paedophile activity on the Internet. In Systematic Approaches to Digital Forensic Engineering, 2005. First International Workshop on, pp. 206–220.

[79] F. A. P. Petitcolas, M. Kutter (2001). Fair Evaluation Methods for Image Watermarking Systems. Journal of Electronic Imaging 9(4):445–455.
[80] J. Platt (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In B. S. D. S. A.J. Smola, P. Bartlett (ed.), Advances in Large Margin Classifiers, pp. 61–74.
[81] A. Qamra, Y. Meng, E. Chang (2005). Enhanced Perceptual Distance Functions and Indexing for Image Replica Recognition. IEEE Transaction on Pattern Analysis and Machine Intelligence 27(3):379–391.
[82] Y. Rui, T. Huang, S. Chang (1999). Image retrieval: current techniques, promising directions and open issues. Journal of Visual Communication and Image Representation 10(4):39–62.