Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

АТЕСТАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Моделі локалізації об’єктів на зображенні у мережевих комп’ютерних системах аналізу зображень
(тема)

Виконав:
студент II курсу, групи СПзм-18-2
Кислинський О.В.
(прізвище, ініціали)

Спеціальність
123 – Комп’ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма
Системне програмування
(повна назва освітньої програми)

Керівник: проф. Смеляков К.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ ______________ Коваленко А.А.
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 – Комп’ютерна інженерія
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри ____________
(підпис)
“____”______________ 20__ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Кислинському Олексію Вадимовичу
(прізвище, ім’я, по батькові)

1. Тема роботи Моделі локалізації об’єктів на зображенні у мережевих комп’ютерних системах аналізу зображень

затверджена наказом по університету від “ 30 ” березня 2020 р. № 43 Стз

2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2020 р.

3. Вхідні дані до роботи
Загальні моделі локалізації і сегментації зображення
Вимоги до об’єктів, моделі та алгоритму
Вимоги до оцінювання ефективності

4. Перелік питань, що потрібно опрацювати в роботі
Аналіз предметної області
Реалізація адаптивних моделі та алгоритму локалізації
Обчислювальний експеримент
Висновки
Аналіз предметної області

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп’ютерних ілюстрацій (слайдів) Демонстраційні матеріали. Плакати – 11 арк. ф. А4

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім’я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз сучасного стану проблеми	31.03.20 – 12.04.20	
2	Постановка задачі дослідження	13.04.20 – 15.04.20	
3	Аналіз літератури	16.04.20 – 20.04.20	
4	Проектування моделі й алгоритмів	21.04.20 – 24.04.20	
5	Реалізація програмної частини	25.04.20 – 28.04.20	
6	Тестування програми та аналіз результатів	29.04.20 – 03.05.20	
7	Оформлення пояснювальної записки	04.05.20 – 11.05.20	
8	Подання атестаційної роботи керівникові	12.05.20 – 14.05.20	
9	Рецензування атестаційної роботи	15.05.20 – 17.05.20	
10	Подання атестаційної роботи до ЕК	18.05.20	

Дата видачі завдання 30 березня 2020 р.

Студент ____________________
(підпис)

Керівник роботи ____________________ проф. Смеляков К.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка атестаційної роботи: 78 с., 47 рис., 0 табл., 1 додаток, 22 джерела.

ЛОКАЛІЗАЦІЯ, СЕГМЕНТАЦІЯ ЗОБРАЖЕННЯ, ГРАНИЦЯ ЗОБРАЖЕННЯ, ДЕТЕКТОР, ЗГОРТКА, МАСКА, МЕТОД ВОДОРОЗДІЛОВ, ОБЧИСЛЮВАЛЬНА ЕФФЕКТІВНІСТЬ.

Метою атестаційної роботи є створення сучасної інформаційної системи для ефективної контурної сегментації і локалізації зображення, моделі якої можуть адаптуватися до особливостей знімків, що обробляються.

У ході виконання атестаційної роботи було проаналізовано існуючі моделі й алгоритми контурної сегментації і локалізації цифрового зображення, розроблено та реалізовано комп’ютерну систему для адаптивної сегментації і локалізації на мові C#, проведено оцінювання якості обробки зображення.

Проведений порівняльний аналіз роботи запропонованого адаптивного алгоритму контурної сегментації і локалізації з аналогами показав суттєві переваги запропонованого алгоритму, які були обумовлені більш високою обчислювальною ефективністю і якістю за рахунок використання інваріантних масок і детекторів.

ABSTRACT

Master’s thesis: 78 pages, 47 figures, 0 tables, 1 appendices, 22 sources.

LOCALIZATION, SEGMENTATION, IMAGE BORDER, DETECTOR, MASK, CONVOLUTION, HYDROCARBON METHOD, COMPUTATIONAL EFFICIENCY.

The purpose of attestation work is to create a modern information system for efficient contour segmentation and localization of digital image, which models can adapt to the peculiarities of the images being processed.

In the course of performance of attestation work, existing models and algorithms of contour segmentation and localization of digital image were analyzed, the computer system for adaptive segmentation of boundaries in C # language was developed and implemented, and an estimation of image processing quality was performed.

The comparative analysis of the proposed adaptive algorithm of contour segmentation and localization with analogues has shown the significant advantages of the proposed algorithm, which were due to higher computational efficiency and quality due to the use of invariant masks and detectors.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN – згорткові нейронні мережі

ГНМ – глибокі нейронні мережі

ВСТУП

В даний час з метою аналізу обстановки на місцевості, з метою геологорозвідки і рішення ряду інших подібних завдань активно використовуються безпілотні літальні апарати. При цьому використання цифрової апаратури дозволяє отримувати і оперативно передавати аерофотознімки з високою роздільною здатністю в центр обробки даних для їх дешифрування [1-3].

Залежно від умов отримання зображення для цілей його поліпшення на етапі попередньої обробки може вирішуватися надзвичайно широкий спектр завдань, основними з яких є завдання поступової корекції, фільтрації шуму, згладжування і підвищення різкості [4, 5].

В даний час практично у всіх додатках, пов’язаних з обробкою цифрових зображень, в більшій або в меншій мірі доводиться вирішувати завдання їх поліпшення [6-8]. В силу наявності більш, чи менш вагомих спотворень знімка. В таких умовах часто виникають проблеми з адекватним сприйняттям, дешифруванням і ефективної сегментацією об’єктів на знімку в сенсі не виявлення слабо контрастних об’єктів; при цьому істотно зростає трудомісткість дешифрування слабо контрастних об’єктів на знімку [1, 9, 10].

При аналізі даних аерофотознімання така ситуація складається не так вже й рідко. В результаті стає неможливим використовувати сучасні системи комп’ютерного зору для аналізу зашумлених слабо контрастних зображень. Стає неможливим оперативно надавати адекватні результати аналізу оброблюваних знімків [11-14].

Подібна ситуація негативним чином відбивається і на можливостях передачі даних по каналах зв’язку. Вся справа в тому, що для зниження часу передачі знімки стискаються [1, 2, 15]. Наявність істотного шуму на зображенні не дозволяє ефективно стискати знімки, а застосування стиснення до знімка, який містить слабо контрастні об’єкти, призводить до додаткової втрати їх контрастності, що неприпустимо [16, 17].

В цілому аналіз сучасного стану питання показує, що за останні кілька десятирічний розроблено воістину величезна кількість методів контурної сегментації і, відповідно, локалізації зображень [1-3, 18, 19].

Разом з тим при такому розмаїтті потрібно витрачати величезний час на вибір і налаштування моделей на особливості оброблюваних знімків. З огляду на можливі варіації умов отримання даних, в такій ситуації дуже складно побудувати уніфіковану інформаційну систему чи технологію адаптивної сегментації і локалізації границь зображень і ліній.

Для вирішення сформульованої проблеми актуально розробити модель і алгоритм контурної сегментації і локалізації цифрового зображення, які інваріантні до найбільш поширених перетворень зображень, а також, які здатні адаптуватися до особливостей об’єктів, що обробляються, для досить широких варіацій умов отримання даних.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ СЕГМЕНТАЦІЇ І ЛОКАЛІЗАЦІЇ ЗОБРАЖЕННЯ

1.1 Детектори об’єктів в глибоких сценах

У теперішній час найбільш поширені моделі, які використовують задля локалізації (детекції) об’єктів на зображенні – це згорткові нейронні мережі (CNN). Тому розглянемо особливості застосування цих мереж.

Нинішні глибокі нейронні мережі досягають неабияких показників у ряді завдань із баченням, що перевершують методи, засновані на штучних функціях. У багатьох випадках ці функції часто зрозумілі та інтерпретовані, у випадку глибоких мереж вони залишаються незрозумілими, яка природа вивченого уявлення і чому воно так добре працює. Звитий нейрон мережа (CNN), що навчається на ImageNet (рис. 1.1), значно перевершує найкращу функцію класичної локалізації. Але що більш дивно, та ж мережа, коли використовується як екстрактор загальних функцій, також є дуже успішною в інших завданнях (не тільки розпізнавання), наприклад, виявлення об’єктів у наборі даних. Ряд робіт було зосереджено на розумінні репрезентації, засвоєної CNN.

Layer	conv1	pool1	conv2	pool2	conv3	conv4	conv5	pool5	fc6	fc7
Units	96	96	256	256	384	384	256	256	4096	4096
Feature	55×55	27×27	27×27	13×13	13×13	13×13	13×13	6×6	1	1

Рисунок 1.1 – Параметри мережевої архітектури, використовувані для ImageNet-CNN та Places-CNN [18].

Під час навчання CNN для розрізнення різних класів об’єктів незрозуміло, яким має бути базове представлення. Об’єкти часто описують за допомогою часткових уявлень, де частини можуть бути спільними для об'єктів, утворюючи розподілений код. Однак, якими вони повинні бути, це і є незрозумілим. Наприклад, можна подумати, що значущими частинами обличчя є рот, два ока і ніс. Однак це просто функціональні частини зі словами, пов'язаними з ними; то деталі об'єкта, важливі для візуального розпізнавання, можуть відрізнятися від цих смислових частин, що ускладнює оцінку ефективності представництва.

Насправді сильна внутрішня конфігурація об'єктів робить визначення того, що є корисною частиною, погано обмежене: алгоритм може знаходити різні та довільні конфігурації деталей, які дають аналогічні показники розпізнавання.

Навчання може класифікувати сцени (тобто класифікувати зображення як офіс, ресторан, вулицю тощо) на основі використання набору даних про місця, що дає можливість вивчити внутрішнє представлення CNN з іншого кута зору, ніж розпізнавання об'єктів.

У випадку сцен уявлення чіткіше. Категорії сцен визначаються об'єктами, вони містять і певною мірою просторову конфігурацію цих об'єктів. Наприклад, важливими частинами спальні є ліжко, тумбочка, світильник, шафа, а також стіни, підлога і стеля. Тому об'єкти представляють розподілений код для сцен (тобто класів об'єктів є поділяється на різні категорії сцен). Важливо, що в сценах є просторова конфігурація об'єктів, хоча компактний, але має значно більшу ступінь свободи. Саме ця вільна просторова залежність робить представлення сцени відмінним від більшості класів об'єктів (більшість класів об'єктів цього не роблять мати слабку взаємодію між частинами). Крім предметів, інші особливості закономірності сцени категорії дозволяють виникнути інші зображення, наприклад, такі як текстури (рис. 1.2).

Основний внесок цього огляду полягає в тому, щоб показати, що виявлення об'єктів виникає всередині CNN, яку навчено розпізнавати/детектувати сцени, навіть більше, ніж під час навчання з ImageNet. Це дивно, адже результати продемонструють, що надійні детектори об’єктів знайдені, хоча, на відміну від ImageNet (рис. 1.3, рис. 1.4).

Рисунок 1.2 – Текстура трави.

Важливо, що поява детекторів об’єктів всередині CNN говорить про те, що єдина мережа може підтримувати розпізнавання на кількох рівнях абстракції (наприклад, краї, текстура, предмети та сцени) без потреби декількох виходів або колекції мереж. Тоді як інші роботи показали, що можна виявляти об’єкти, застосовуючи мережу кілька разів у різних місцях, або фокусування уваги, або шляхом сегментації, буде показано, що одна і та ж мережа може робити і локалізацію об’єктів, і розпізнавання сцени в єдиному ланцюгу.

Якщо спеціально не оговорене інше, будемо розглядати мережі ImageNet-CNN і Places-CNN.

Нещодавно конволюційні нейронні мережі отримали приголомшливі показники щодо об’єктної класифікації і локалізації, та класифікації сцен.

ImageNet-CNN від Jia (2013) навчається на 1,3 мільйонb зображень з 1000 об’єктних категорій ImageNet (ILSVRC, 2012 р.) Та досягає першої точності 57,4%. З тією ж мережевою архітектурою Places-CNN проходить 2,4 мільйона зображень із 205 категорій сцен бази даних місць і досягає першої точності 50,0%.

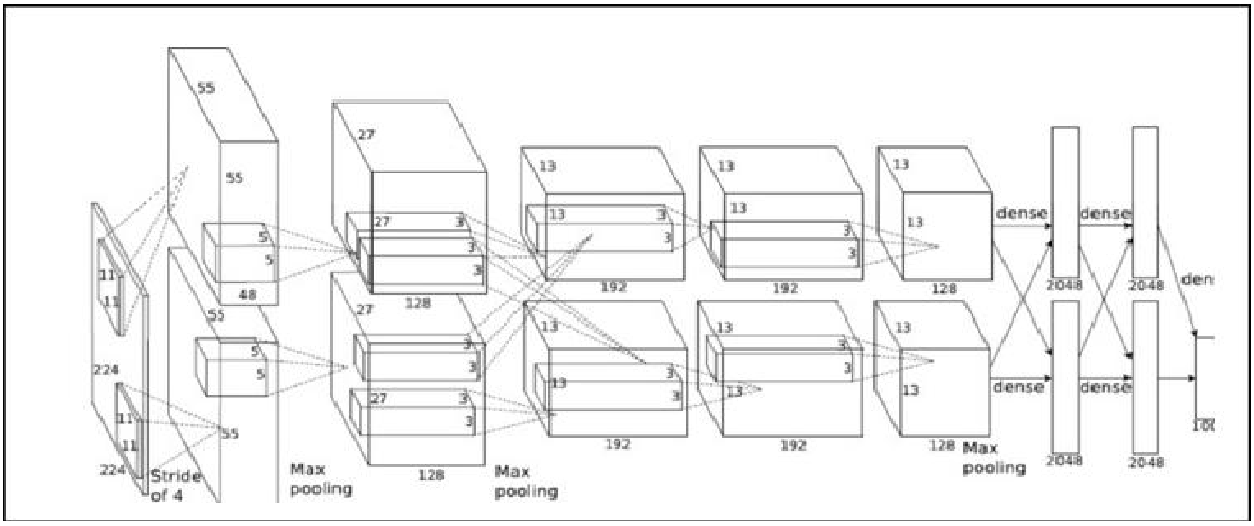


Рисунок 1.3 – Модель ImageNet-CNN [18].

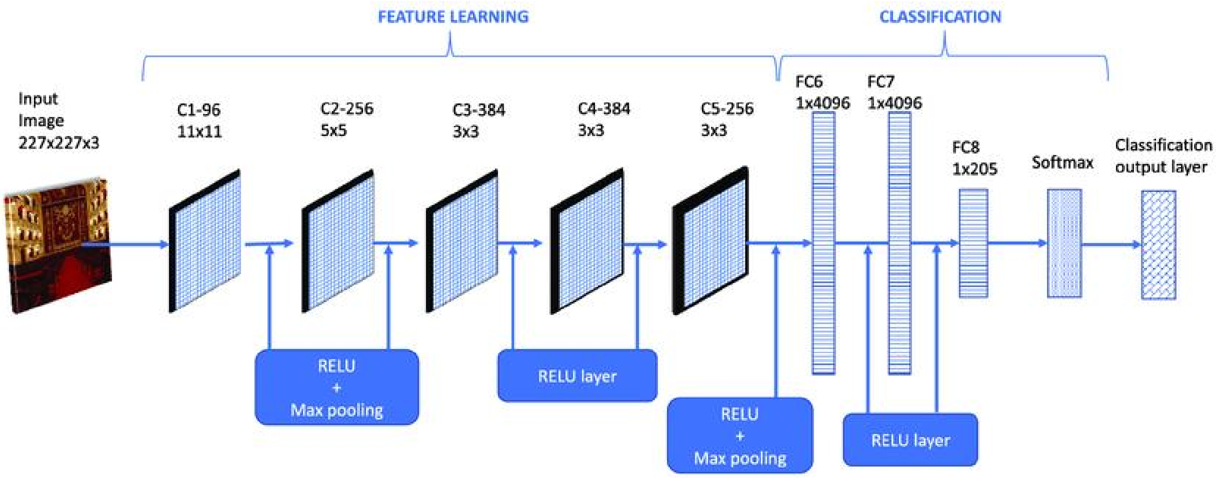


Рисунок 1.4 – Модель Places-CNN [18].

Мережева архітектура, що використовується для обох CNN, узагальнено в таблиці (рис. 1.1).

Обидві мережі навчаються з нуля, використовуючи лише вказаний набір даних.

Глибокі функції від Places-CNN, як правило, краще справляються із завданнями розпізнавання сцени порівняно з функціями ImageNet-CNN.

Наприклад, порівняно з Places-CNN це досягає 50,0% за класифікацією сцен, ImageNet-CNN поєднується лише з лінійним SVM досягає 40,8% на тому ж тестовому наборі, що ілюструє важливість наявності даних, орієнтованих на сцени.

Щоб ще більше підкреслити різницю уявлень, ми проводимо простий експеримент для виявлення відмінності в типі зображень, бажаних на різних шарах кожної мережі: ми створюємо набір 200 тис. зображень із приблизно рівним розподілом зображень, орієнтованих на сцену та об’єктно-орієнтовані, та запускаємо їх через обидві мережі, записуючи активації на кожному рівні (і зберігаючи їх).

Для кожного шару ми отримати топ-100 зображень із найбільшою середньою активацією (сума для всіх просторових місць для заданого шару). На рисунку 1.5 показано 3 верхніх зображення для кожного шару [18]. Ми спостерігаємо, що більш ранні шари такі, як pool1 і pool2 віддають перевагу подібним зображенням для обох мереж, тоді як пізніші шари, як правило, більше спеціалізуються на конкретній задачі категоризації сцен або об’єктів.

Рисунок 1.5 – Топ-3 зображення, що дають найбільшу активацію одиниць у кожному шарі ImageNet-CNN (вгорі) та Places-CNN (знизу).

Для шару пулу2, 55% і 47% топ-100 зображень належать до набору даних ImageNet для ImageNet-CNN та Places-CNN. Починаючи від шару conv4 ми спостерігаємо істотну різницю в кількості топ-100, що належать кожному набору даних, що відповідає кожній мережі. Для fc7 ми спостерігаємо, що 78% та 24% найпопулярніших 100 зображень належать до набору даних ImageNet для ImageNet-CNN та Places-CNN відповідно, що ілюструє чітку упередженість у кожній мережі.

Надалі ми додатково досліджуємо відмінності між цими мережами та фокусуємось щодо кращого розуміння природи представлення, засвоєного Places-CNN під час виконання класифікації / локалізації сцени.

Виконання розпізнавання сцен за допомогою Places-CNN є вражаючою, враховуючи складність завдання.

Спрощення зображень – це добре відома стратегія перевірки людського розпізнавання. Наприклад, можна видалити інформацію з зображення, щоб перевірити, чи є вона діагностичною чи ні. Аналогічну процедуру також застосував Танака (1993) щоб зрозуміти сприйнятливі поля складних клітин у нижній скроневій корі. Натхненна цими підходами, наша ідея така: задане зображення, яке правильно класифіковане мережею ми хочемо спростити таким чином, щоб воно зберігало як мінімум візуальної інформації; можливо, все ще маючи високий бал класифікації для тієї ж категорії.

Це спрощене зображення (з назвою мінімальне зображення) дозволить нам виділити елементи, що призводять до високих класифікаційний. Для цього ми маніпулюємо зображеннями в градієнтному просторі, як зазвичай робиться в комп’ютерній графіці. Ми досліджуємо два різних підходи, які описані нижче. У першому підході, задавши зображення, ми створюємо сегментацію країв та областей та видаляємо сегменти з зображення ітеративно. При кожній ітерації ми видаляємо сегмент, який виробляє найменше зниження правильної оцінки класифікації, і ми робимо це, поки зображення не буде неправильним. Наприкінці ми отримуємо подання оригінального зображення, яке містить приблизно, мінімальну кількість інформації, яка необхідна мережі для правильного розпізнавання категорії сцени. Спираючись на попередні результати, ми висунули гіпотезу, що для

Places-CNN деякі об’єкти мають вирішальне значення для розпізнавання сцен. Це надихнуло наш другий підхід: ми формуємо мінімальні зображення, використовуючи повністю анотований набір зображень SUN Database, замість автоматичної сегментації. Ми дотримуємось тієї ж процедури, що і перший підхід, використовуючи сегменти об’єкта.

Проведені експерименти (рис. 1.6, рис. 1.7) призвели до цікавих спостережень: для спалень мінімальне уявлення зберегло ліжко в 87% випадків. Інші предмети, що зберігаються в спальнях, були настінні (28%) та віконні (21%). Для художньої галереї мінімальні зображення зображень містили картини (81%) та картини (58%); в парках розваг, каруселі (75%), їзді (64%) та американських гірках (50%); у книгарні, книжковій шафі (96%), книги (68%) та полиці (67%). Ці результати говорять про те, що виявлення об’єктів є важливою частиною представлення, побудованого мережею для отримання дискримінаційної інформації для сцени класифікація.

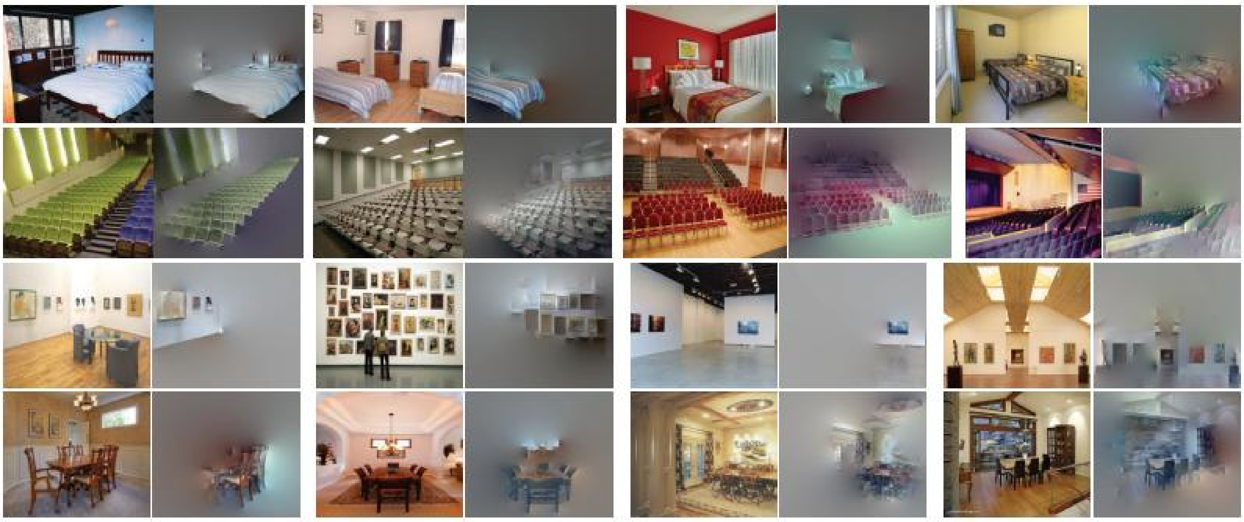


Рисунок 1.6 – Кожна пара зображень показує оригінальне зображення (зліва) та спрощене зображення (праворуч), яке отримується Places-CNN як та сама категорія сцени, що і оригінальне зображення. Зверху вниз, чотири ряди показують різні категорії сцен: спальня, аудиторія, галерея та їдальня.

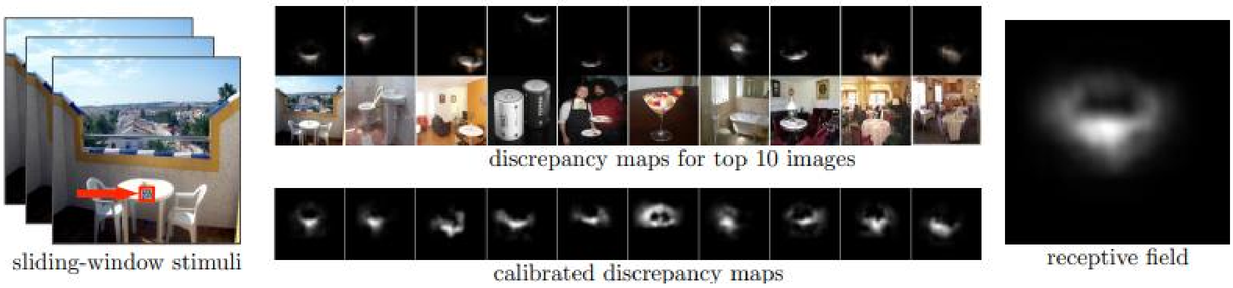


Рисунок 1.7 – Трубопровід для оцінки кожного блоку. Кожен подразник розсувного вікна містить невелике рандомізоване виправлення (приклад, зазначений червоною стрілкою) в різних просторових місцях. Порівнюючи реакцію активації подразників ковзання з реакцією активації оригіналу зображення, ми отримуємо карту невідповідності для кожного зображення (середній верхній). Підсумовуючи відкалібровані карти невідповідності (середня нижня частина) для зображень з найбільш високим рейтингом, ми отримуємо фактичну частоту цього блоку (справа).

Тепер розглянемо питання візуалізації рецептивних полів і деякі аспекти активізації. Будемо досліджувати форму та розміри сприйнятливих полів (ВЧ) різних одиниць у CNN. Хоча теоретичні розміри ВЧ можна обчислити з огляду на мережеву архітектуру, цікавить фактичний чи емпіричний розмір РФ. Ми очікуємо, що емпіричні РФ бути кращою локалізацією та більш представницькою інформацією, яку вони збирають, ніж теоретичною, що дозволяє нам краще зрозуміти, що дізнається кожен підрозділ CNN.

Таким чином, пропонуємо керований даними підхід для оцінки вивченого РФ кожного блоку в кожному шарі. Це є простіший шлях, ніж деконволюційний метод візуалізації мережі і може легко розширюється для візуалізації будь-яких вивчених CNN.

Процедура оцінки ВЧ заданої одиниці, як показано на рис. 1.7, полягає в наступному. Як вхід, ми використовуйте набір зображень із 200 тис. зображень із приблизно рівним розподілом сцен та об’єктів. Потім вибираємо верхні K-зображень із найвищими активаціями для даного блоку.

Для кожного з к зображень ми хочемо зараз точно визначити, до яких областей зображення веде висока активація блоку. Для цього ми повторюємо кожне зображення багато разів невеликими випадковими фрагментами (патчі зображення розміром 11 × 11) у різних місцях зображення.

Зокрема, ми генеруємо фрагменти в щільній сітці з кроком 3. Це призводить до отримання приблизно 5000 фрагментарних зображень на оригінальне зображення.

Тепер ми подаємо всі закриті зображення в одну мережу і записуємо зміну активації як порівняно з використанням оригінального зображення. Якщо є велика невідповідність, ми знаємо, що даний патч важливий і навпаки. Це дозволяє нам побудувати карту розбіжностей для кожного зображення.

Нарешті, для закріплення інформації з к зображень ми зосереджуємо карту розбіжностей навколо просторового розташування блоку, яке спричинило максимальну активацію для даного зображення. Тоді ми отримуємо середнє значення перецентрованих карт невідповідності для отримання остаточного РФ. На рисунку 1.8 візуалізовані частотні сигнали для одиниць із 4-х різних шарів Places-CNN та ImageNetCNN, а також їх найвищих областей активації в межах РФ.

Ми спостерігаємо це як шари заглиблюючись, розмір РФ поступово збільшується, а області активації стають все більш семантично змістовними. Далі, як показано на рисунку 1.9, ми використовуємо РФ для сегментації зображень за допомогою карт функцій різних одиниць. Нарешті, на рисунку 1.10 наведено порівняння теоретичного та емпіричного розміру РФ для різних шарів. Як і очікувалося, фактичний розмір РФ набагато менший, ніж теоретичний розмір, особливо в пізніших шарах.

В цілому цей аналіз дозволяє нам краще зрозуміти кожну одиницю фокусуючись точно на важливих регіонах кожного зображення.

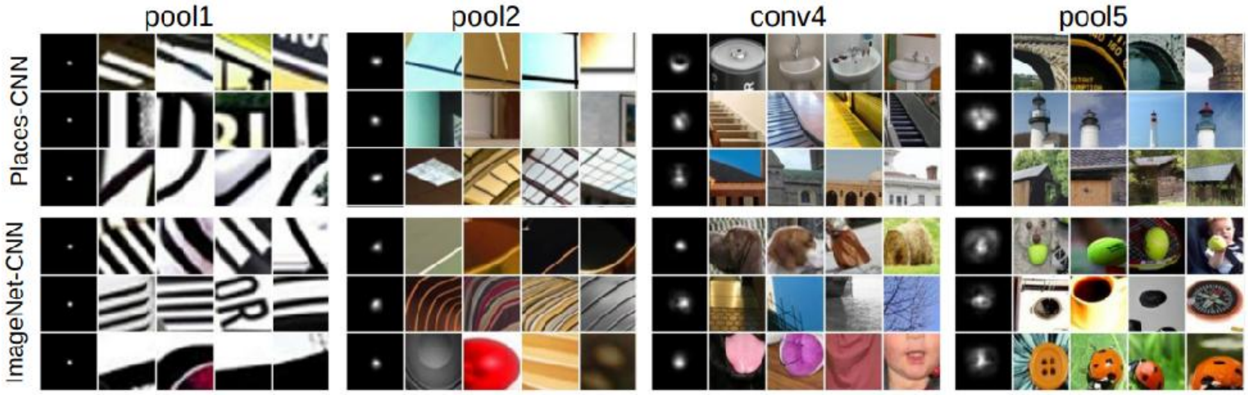


Рисунок 1.8 – РФ 3 одиниць шарів пул1, пул2, конв4 та пул5 відповідно для ImageNet та Places-CNNs, а також патчі зображень, що відповідають верхнім областям активації всередині РФ.

Рисунок 1.9 – Сегментація на основі РФ. Кожен рядок показує 4 найвпевненіших зображення для блоку.

	pool1	pool2	conv3	conv4	pool5
Theoretic size	19	67	99	131	195
Places-CNN actual size	17.8± 1.6	37.4± 5.9	52.1±10.6	60.0± 13.7	72.0± 20.0
ImageNet-CNN actual size	17.9± 1.6	36.7± 5.4	51.1±9.9	60.4± 16.0	70.3± 21.6

Рисунок 1.10 – Порівняння теоретичних та емпіричних розмірів РФ для ImageNet та Places-CNN на різних шарах. Зауважимо, що РФ прийнято будувати як квадратну, а розміри, наведені нижче, - це довжина кожної сторони цього квадрата в пікселях.

Розглянемо ідею ідентифікації семантики внутрішніх одиниць (в нутрощі зображення). Вище ми знайшли точні частоти одиниць і помітили, що області активації мають тенденцію – вони стають більш семантично значущими із збільшенням глибини шарів.

Тепер потрібно зрозуміти та кількісно оцінити точну семантику, засвоєну кожною одиницею.

Для цього було використано сервіс Amazon Mechanical Turk (AMT) для визначення спільної тема або концепції, що існує між верхніми оцінками сегментів для кожної одиниці. Ми очікуємо теги, надані наївними анотаторами, щоб зменшити упередження.

Зокрема, ми ділимо завдання на три основні етапи, як показано на рисунку 1.11. Ми показуємо експертам топ-60 сегментованих зображень, які найбільш сильно активують один блок, і ми просимо їх (1) визначити концепцію, або семантичну тему, задану набором з 60 зображень, наприклад, автомобіль, синій, вертикальні лінії тощо, (2) позначають набір зображень, які не підпадають під цю тему, і (3) класифікувати поняття, передбачене в (1), до одного з 6 семантичних груп, починаючи від низького рівня до високого рівня: прості елементи та кольори (наприклад, горизонтальні лінії, синій), матеріали та текстури (наприклад, дерево, квадратна сітка), області та поверхні (наприклад, дорога, трава), деталі предмета (наприклад, голова, нога), предмети (наприклад, автомобіль, людина) та сцени (наприклад, кухня, коридор).

Усе це дозволяє отримати як семантичну інформацію для кожної одиниці, так і рівень абстракції, що забезпечується маркованою концепцією.

Щоб забезпечити високу якість анотації, ми включили 3 зображення з високими негативними балами, які експерти повинні були визначити як негативні, щоб подати завдання.

На рисунку 1.12 показаний деякий приклад анотації експертів. Для кожної одиниці ми вимірюємо її точність як відсоток зображень, які були обрані як відповідні маркованій концепції.

На рисунку 1.13.а побудовано середню точність для ImageNet та Places-CNN для кожного шару, а також розподіл категорій концепцій для ImageNet-CNN та Places-CNN на кожному шарі. Для цього сюжету ми розглядаємо лише одиниці, які мали точність вище 75%, як це передбачено.

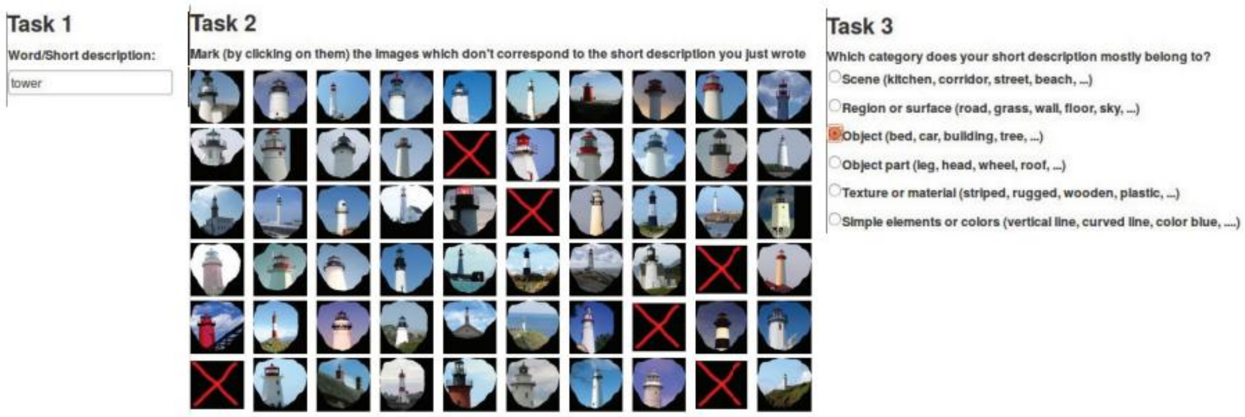
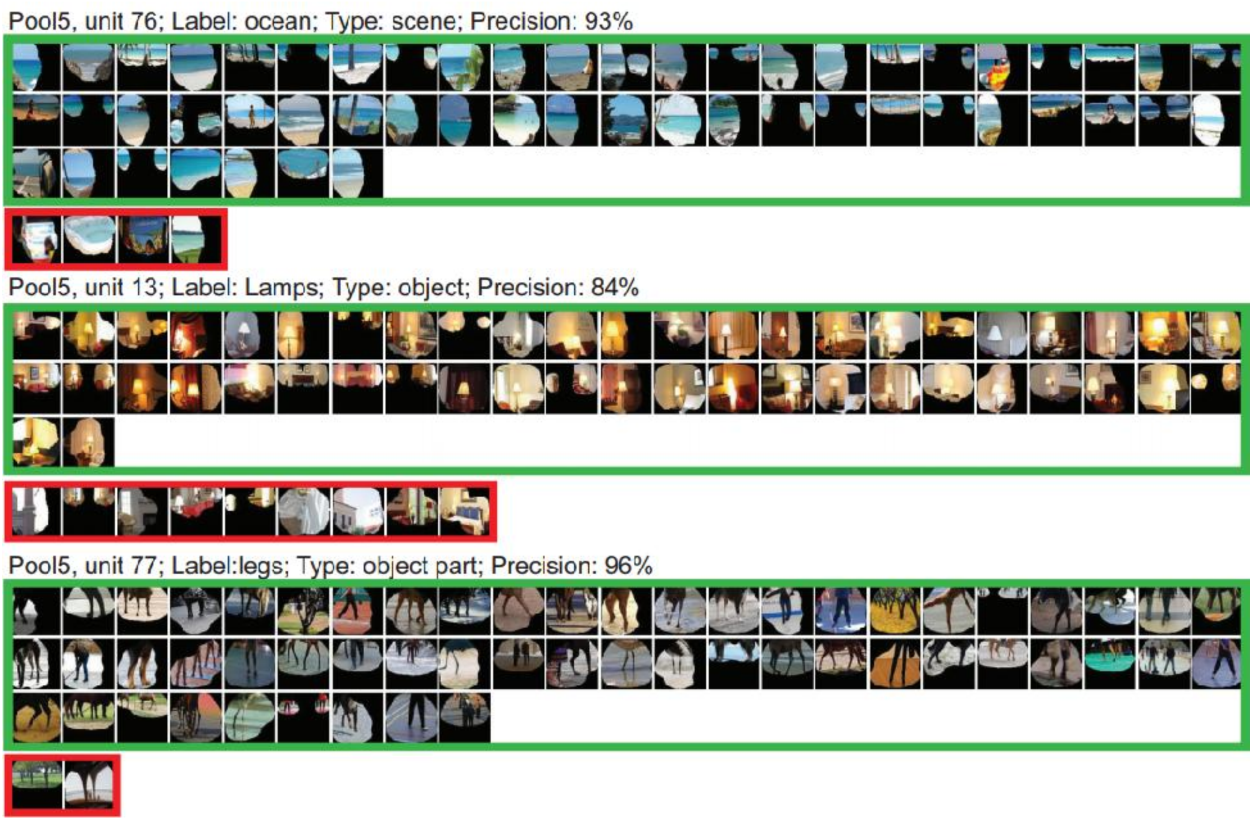
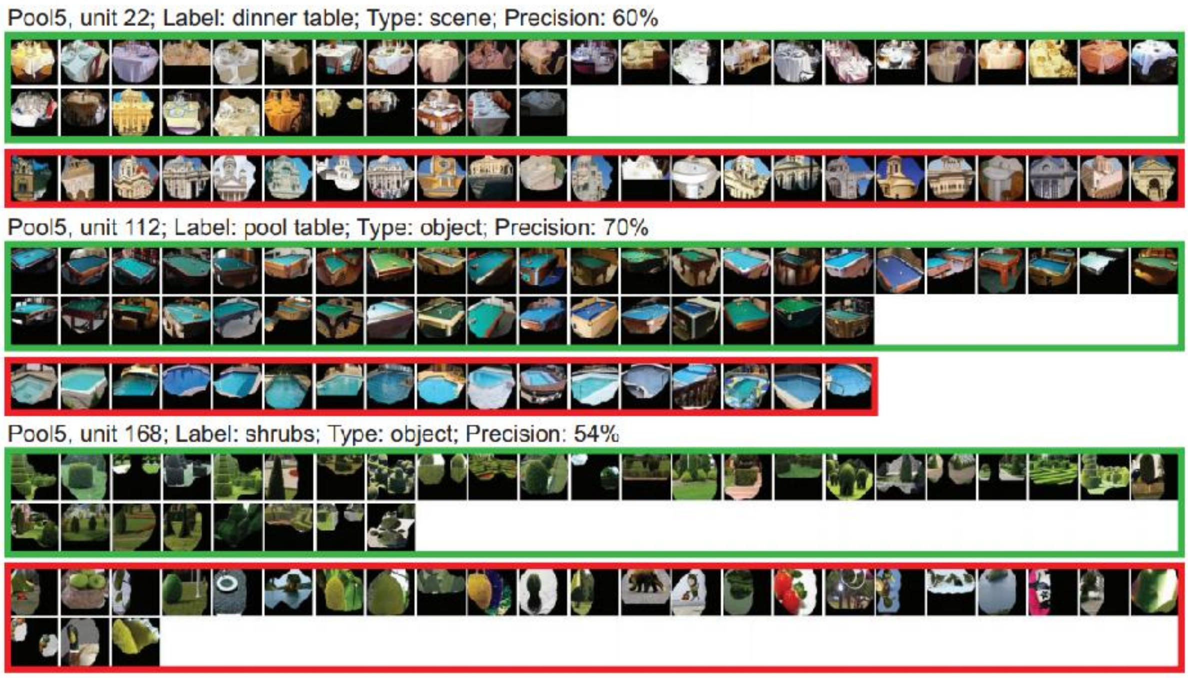


Рисунок 1.11 – Інтерфейс AMT для анотації концепту одиниці. У кожній примітці є три завдання.



а)



б)

Рисунок 1.12 – Приклади анотацій одиниць, надані експертами AMT для 6 одиниць від пулу5 в Places-CNN. На кожній одиниці на рисунку показана мітка, надана працівником, тип етикетки, зображення, вибрано як відповідне поняттю (зелене поле) та зображення, позначені як неправильні (червоне поле). Точність – відсоток правильних зображень. Три кращі блоки мають високу продуктивність, нижня трійка має низьку продуктивність (<75%).

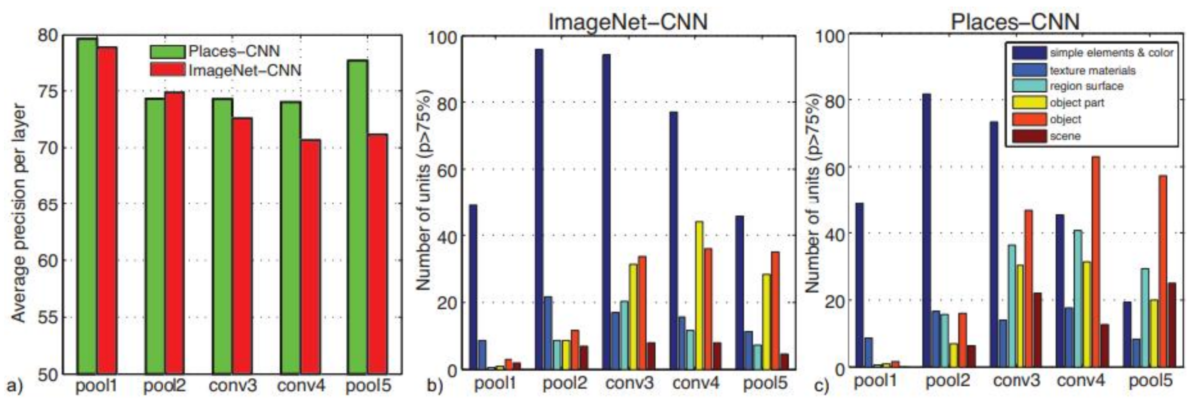


Рисунок 1.13 – (a) Середня точність всіх одиниць у кожному шарі для обох мереж, (b) та (c) показують кількість одиниць, що забезпечують різні рівні семантики для ImageNet-CNN та Places-CNN відповідно.

Близько 60% одиниць на кожному шарі, де вище цього порогу. Для обох мереж, одиниці на ранніх шарах (pool1, pool2) мають більше одиниць, що реагують на прості елементи та кольори, в той час як ті, які є на більш пізніх шарах (conv4, pool5), мають більш високу семантику (більш чутливі) до предметів і сцен).

Крім того, ми спостерігаємо, що conv4 та pool5 одиниць у Places-CNN мають більш високі коефіцієнти семантики високого рівня порівняно з одиницями в ImageNet-CNN.

На рисунку 1.14 наведена інша форма візуалізації тих же даних, що і на рис. 1.13. Цей сюжет краще розкривається якщо виникають різні рівні абстракції в різних шарах обох мереж. Вертикальна вісь вказує відсоток одиниць у кожному шарі, присвоєний кожній категорії концепції.

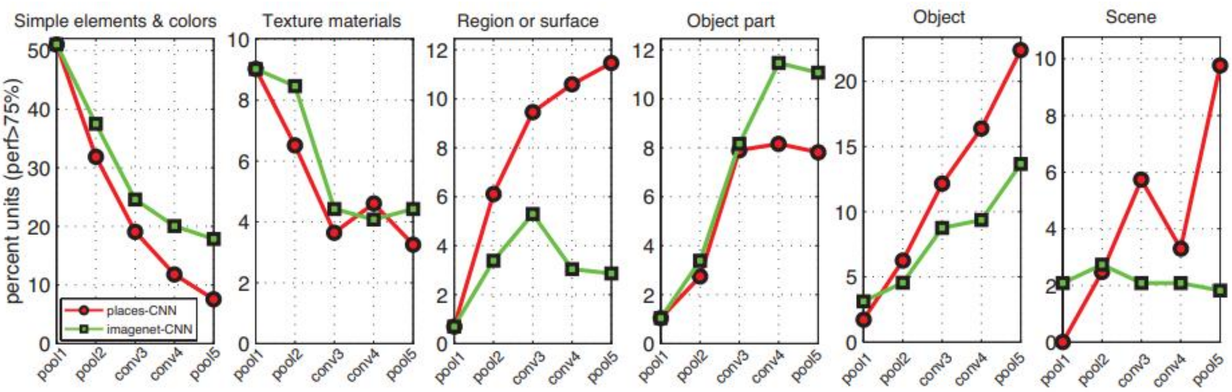


Рисунок 1.14 – Розподіл смислових типів, знайдених для всіх одиниць в обох мережах. Зліва направо, кожен графік відповідає розподілу одиниць у кожному шарі, призначених простим елементам або кольорам, текстури або матеріали, регіони або поверхні, деталі об’єктів, предмети та сцени. Вертикальна вісь – це відсоток одиниць із кожним шаром, присвоєним кожному типу концепції.

ImageNet-CNN має більше одиниць, налаштованих на прості елементи та кольори, ніж Places-CNN, тоді як Places-CNN має більше предметів та сцен. ImageNet-CNN має більше одиниць, налаштованих на деталі об’єкта (максимум навколо conv4). Цікаво відзначити, що Places-CNN виявляє більше об’єктів, ніж ImageNet-CNN, незважаючи на це.

Зараз розглянемо аспект появи (виходу, чи детектування) об’єкту. Як було показано раніше, велика кількість одиниць у пулі 5 присвячена виявленню об’єктів та областей сцен.

Але які категорії знайдені? Чи кожна категорія відображена в одне ціле або існує кілька одиниць для кожного класу об’єктів? Чи можемо ми реально використовувати цю інформацію для сегментації сцени? Які спільні класи ми маємо?

Щоб відповісти на питання, чому певні об’єкти виходять із пулу5, ми протестували ImageNet-CNN та Places-CNN на повністю анотованих зображеннях із бази даних SUN. Ця база даних містить 8220 повністю анотованих зображень із тих самих 205 категорій місць, які використовували для тренувань Places-CNN.

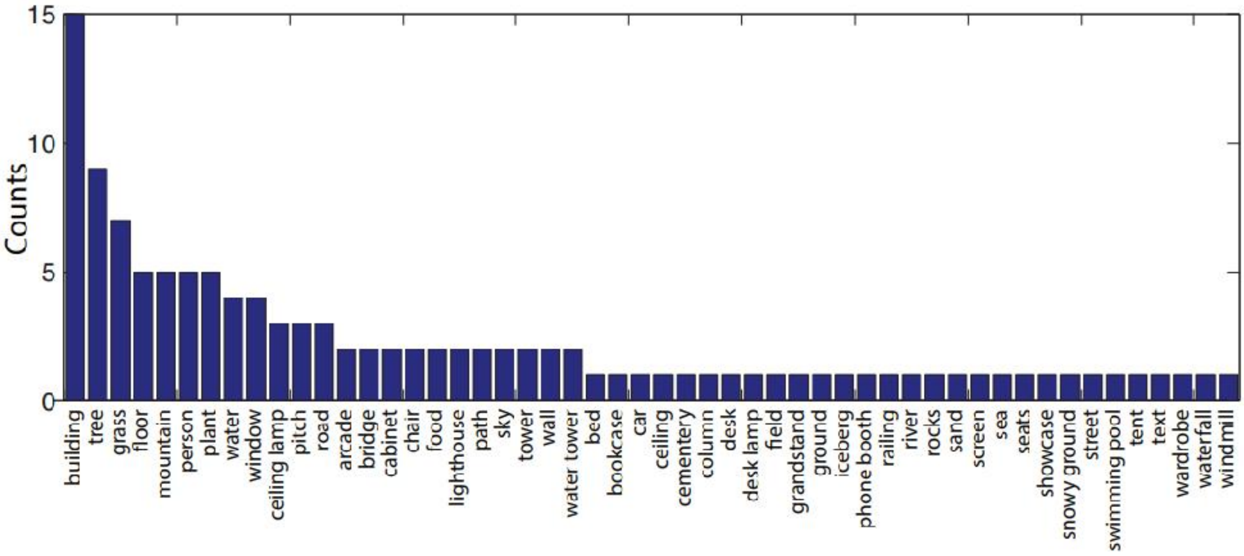
Об’єктними класами для категоризації сцен є те, якими є природні частоти в об’єктах зображення. Для цього дослідження відобразили теги, надані експертами AMT, до категорій SUN.

На рисунку 1.15.а показано розподіл об’єктів, знайдених у пулі5 Places-CNN. Деякі об’єкти виявляються кількома одиницями. Наприклад, існує 15 одиниць, які виявляють будівлі. 11 показані деякі одиниці від групи Places-CNN, згруповані за типом класу об’єктів, який вони, схоже, виявляють.

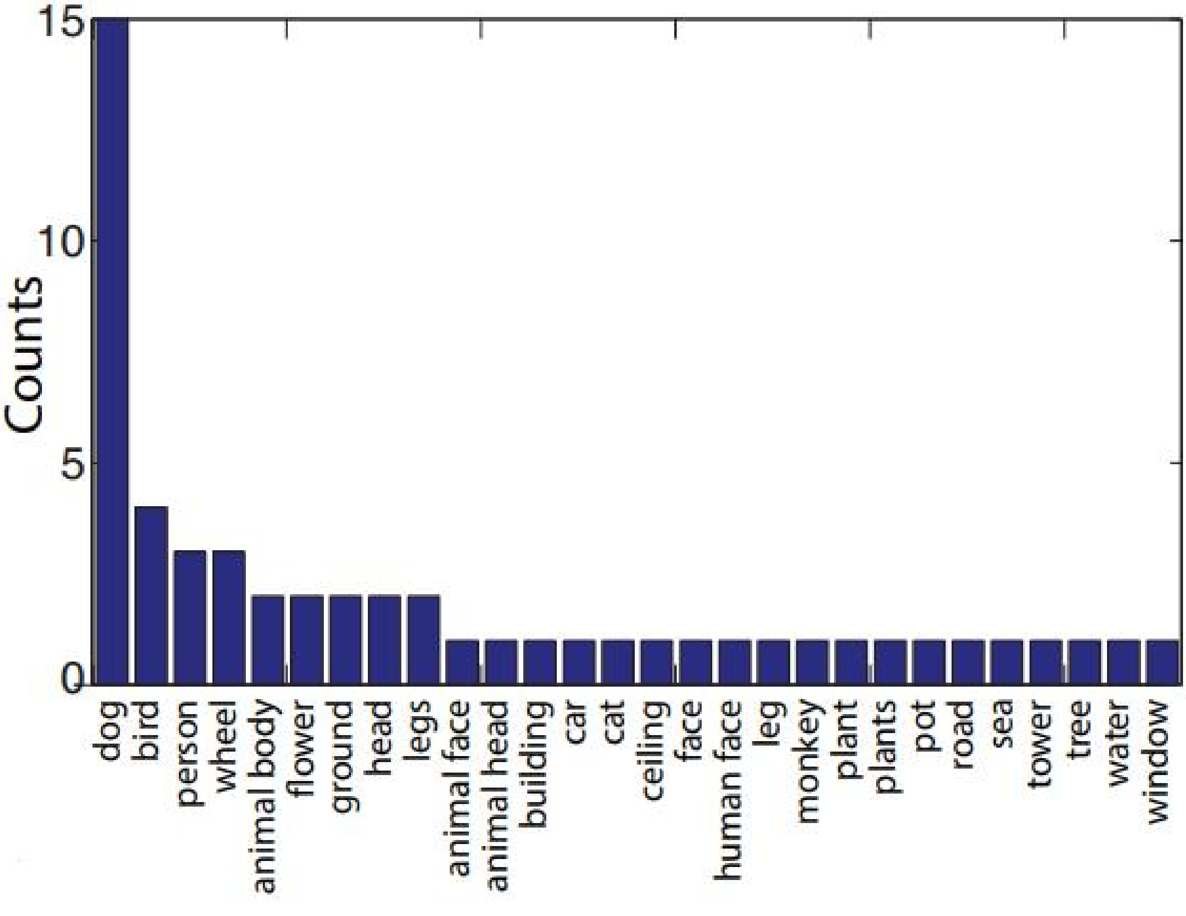
У кожному рядку відображається знак п’ять найкращих зображень для певного блоку, які дають найсильніші активації.

Показана сегментація області зображення, для яких одиниця перевищує певний поріг. Кожна одиниця, здається, є вибіркові до конкретного вигляду об’єкта.

Наприклад, є 6 одиниць, які виявляють лампи, кожен блок виявлення певного типу лампи, що забезпечує більш дрібну дискримінацію; є 9 одиниць вибіркові для людей, кожен налаштований на різні масштаби.



а)



б)

Рисунок 1.15 – Кількість об’єктів-одиниць CNN, що виявляють кожен клас об’єктів для (a) Places-CNN та (b) ImageNet-CNN.

На рисунку 1.15.б показано розподіл об’єктів, знайдених у пулі5 ImageNet-CNN. ImageNet має чисельність тварин серед присутніх категорій: у ImageNet-CNN з 256 одиниць у pool5, є 15 одиниць, присвячених виявленню собак та ще кілька детектуючих частин собак (тіла, ноги, ...).

Категорії, знайдені у pool5, як правило, відповідають цільовим категоріям у ImageNet.

Чому виникають ці об’єкти? Одна з можливостей полягає в тому, що об’єкти, які з’являються в пулі5, відповідають до найчастіших у базі даних.

Ще один цікавий експеримент наведено на рисунку 1.16.

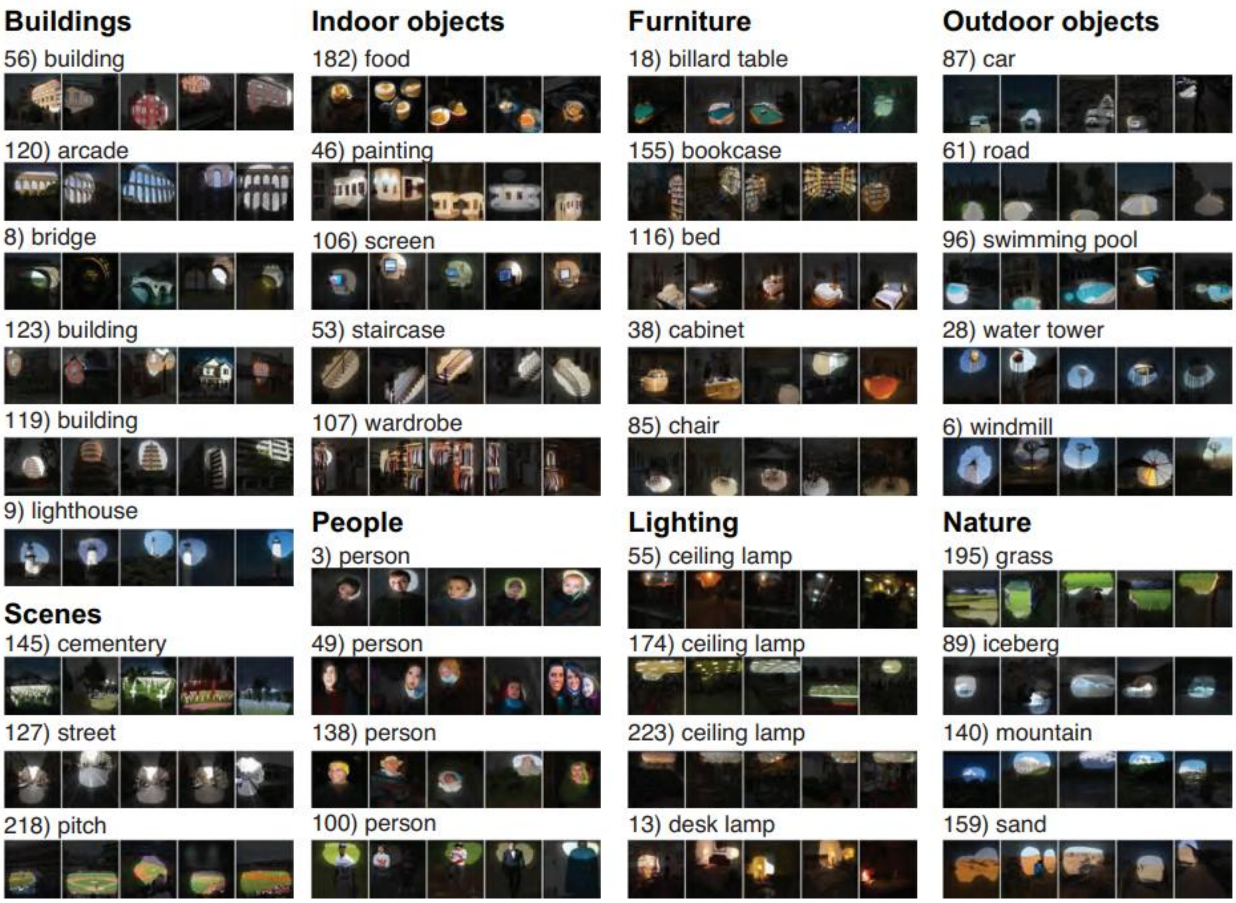


Рисунок 1.16 – Сегментації з використанням пулу5 одиниць від Places-CNN. Багато класів закодовано декількома одиницями, що охоплюють об’єкти різної зовнішності. Кожен рядок показує 5 найбільш впевнених зображень для кожної одиниці (класу). Число представляє номер одиниці в пулі5.

На рисунку 1.17.а показано відсортований розподіл підрахунків об’єктів в базі даних SUN, яка відповідає закону Zipf. На рис. 1.17.б показано кількість одиниць, знайдених у пулі5 для кожного класу об’єктів (те саме сортування, як на рис. 1.17.а). Кореляція між частотою об’єкта в частоті баз даних та об’єктів, виявлена одиницями у пулі5, становить 0,54.

Інша можливість, що об’єкти, що з’являються, - це об’єкти, які дозволяють розрізняти категорії сцен. До вимірювання набору об’єктів, що дискримінують, ми використовували основне положення в базі даних SUN для вимірювання, ефективність класифікації, що досягається кожним класом об’єктів для класифікації сцен.

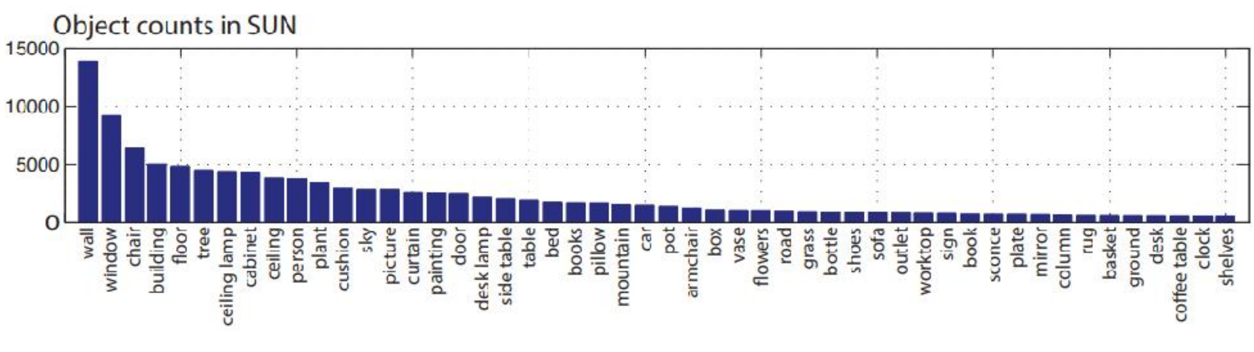
Тоді рахуємо скільки разів кожен клас об’єктів видається найбільш інформативним. Це вимірює кількість категорії сцен, для яких певний клас об’єктів є найбільш корисним. Підрахунки показані на рисунку 1.17(с).

Звернімо увагу на подібність між рисунком 1.17(б) та рисунком 1.17(с). Кореляція дорівнює 0,84, що вказує на те, що мережа автоматично визначає найбільш дискримінаційні категорії об’єктів.

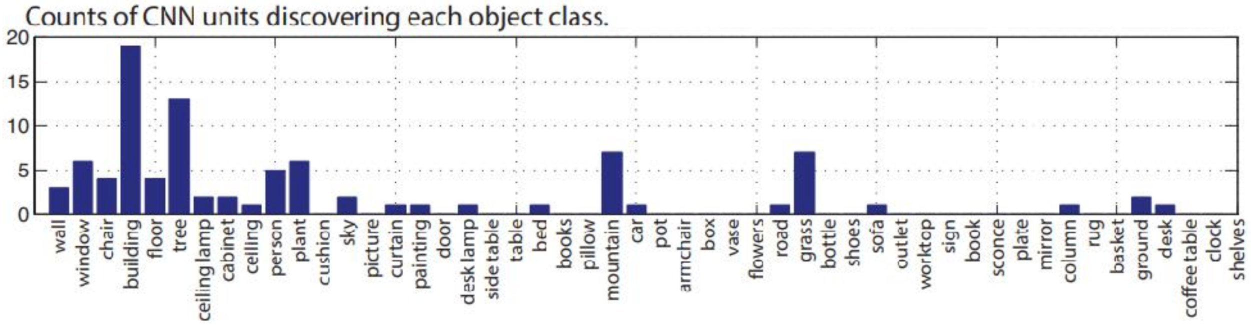
Зауважмо, що 115 пулів у пулі5 Places-CNN не виявляють об’єктів. Це може бути пов’язано з неповним навчанням або доповненням текстур.

Тому, хоча об’єкти здаються ключовою частиною уявлення, яке вивчає мережа, ми не виключаємо використання інших уявлень, що використовуються в поєднанні з об’єктами.

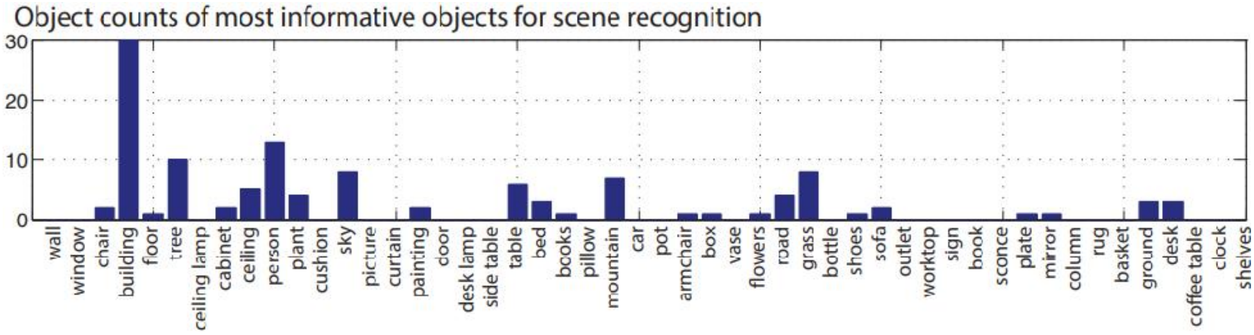
Для того, щоб якість детектування об’єктів була максимальною.



а)



б)



в)

Рисунок 1.17 – (а) Частота об’єктів у SUN (показано лише топ-50 об’єктів), (б) Кількість об’єктів, виявлених басейном5 в Places-CNN і (в) Частота найбільш інформативних об’єктів для класифікації сцен.

Тепер розглянемо питання локалізації об’єкту на внутрішніх шарах. Мережа Places-CNN навчається робити класифікацію сцен, використовуючи результат останнього рівня логістичної регресії та досягаючи сучасних показників. З проведеного вище аналізу видно, що багато підрозділів у внутрішніх шарах можуть виконувати інтерпретаційну локалізацію об’єкта.

Таким чином, ми могли б використовувати цю єдину Places-CNN з анотацією одиниць для розпізнавання сцени та локалізації об’єкта в одному циклі обробки. На рисунку 1.18 показаний приклад виведення різних шарів Places-CNN з використанням тегів. Обмежувальні ящики показані навколо областей, де є кожен блок активований в межах своєї РФ вище певного порогу.

На рисунку 1.19 наведено ефективність сегментації об’єктів, виявлених у пулі5, використовуючи базу даних SUN. Продуктивність багатьох підрозділів дуже висока, що підтверджує що вони дійсно виявляють ці об’єкти-класи, незважаючи на те, що пройшли навчання для класифікації сцен.

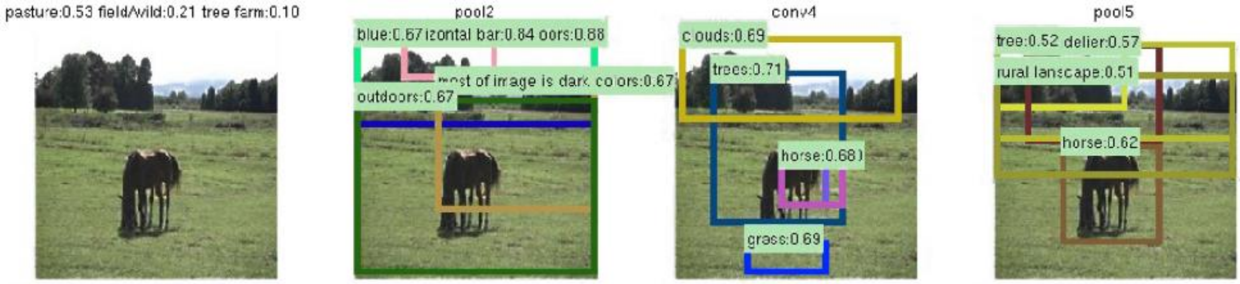
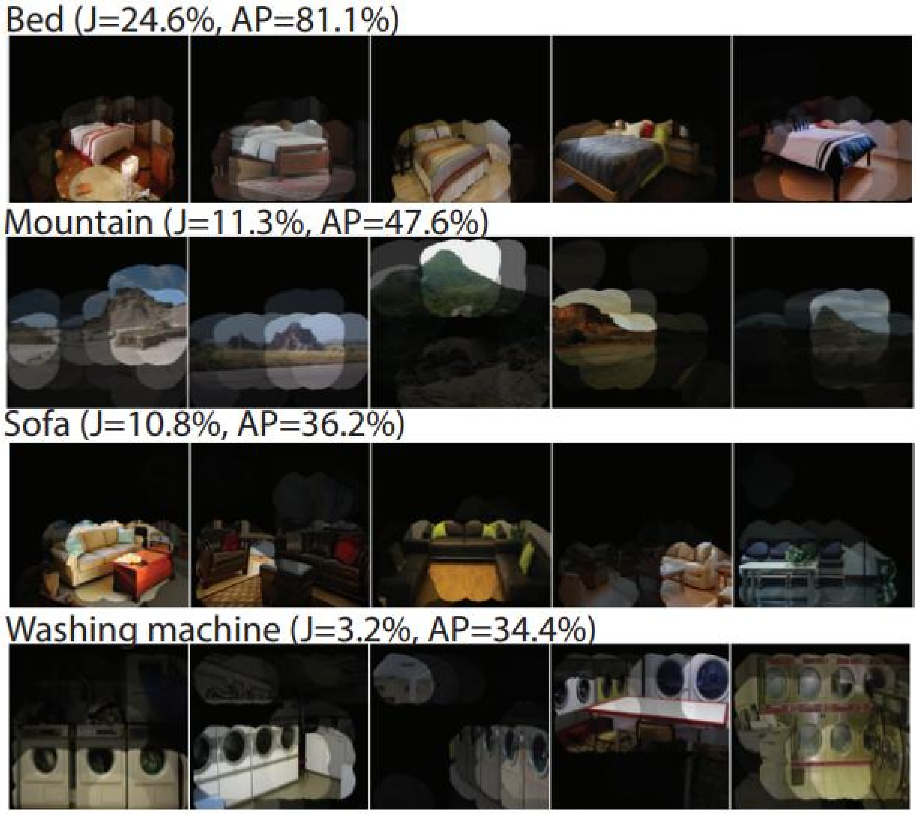
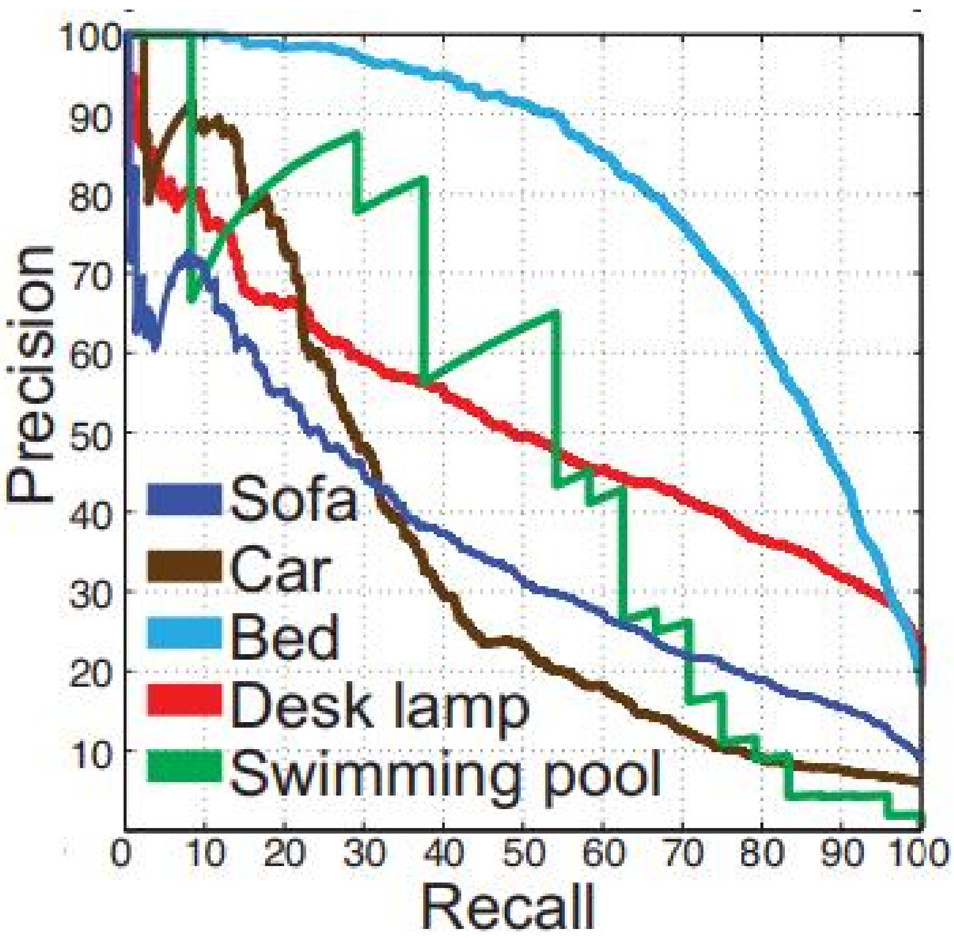


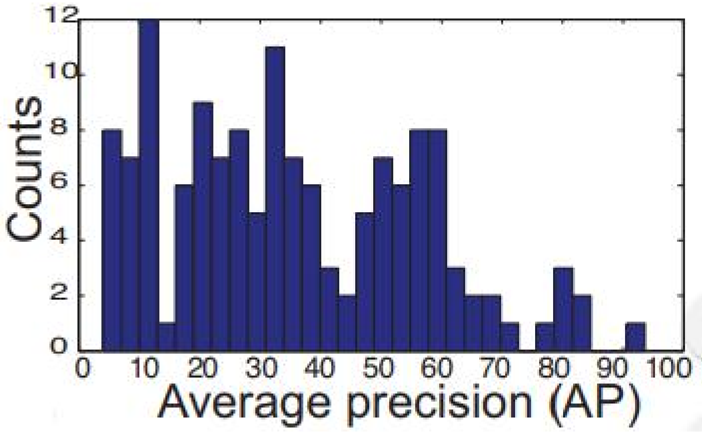
Рисунок 1.18 – Інтерпретація рисунка різними шарами Places-CNN з використанням наданих тегів. Перший показує кінцевий рівень випуску Places-CNN. Інші три результати виявлення разом із достовірністю на основі активації одиниць та семантичних тегів.


Bed (J=24.6%, AP=81.1%)
Mountain (J=11.3%, AP=47.6%)
Sofa (J=10.8%, AP=36.2%)
Washing machine (J=3.2%, AP=34.4%)

а)


Precision
Recall
Sofa
Car
Bed
Desk lamp
Swimming pool
0
10
20
30
40
50
60
70
80
90
100

б)



в)

Рисунок 1.19 – (а) Сегментація зображень із бази даних SUN за допомогою пулу5 Places-CNN (J =Індекс сегментації Жакарда, AP = середня точність-відкликання); (б) Криві точності відкликання для деяких виявлені предмети; (в) Гістограма AP для всіх виявлених об’єктних класів.

Аналіз отриманих даних показує, що ефективні детектори об’єктів з’являються в результаті навчання класифікувати категорії сцен, показуючи, що одна мережа може підтримувати розпізнавання на кількох рівнях абстракції (наприклад, ребра, текстури, об’єкти та сцени), не потребуючи кількох виходів чи мереж.

1.2 Швидкий детектор YOLO

Так, виявлення (детектування) об’єктів є однією з найбільш складних проблем у комп’ютерному зорі. Мета виявлення об’єктів – локалізувати різні об’єкти на сцені та призначити мітки об’єктам.

Останнім часом глибокі нейронні мережі (ГНМ) показали свою перевагу та продуктивність у наборі різних доданків, де об’єктом виявлення є одна з ключових областей на зображенні; ГНМ значно перевершили існуючі підходи. Зокрема, ряд методів, заснованих на конволюційній нейромережі (CNN) продемонстрував істотні переваги для досягнення найсучасніших показників виявлення об’єктів. Наприклад, у підході Region-CNN (R-CNN), CNN-архітектура використовується для генерації пропозицій з обмежувальним вікном у зображення замість розсувного вікна, а отже, класифікатор виконує класифікацію лише на пропозиціях з обмежувальним вікном. Хоча R-CNN здатна виробляти сучасну точність, всю процедуру повільно та важко оптимізувати, оскільки кожен компонент повинен тренуватися індивідуально (рис. 1.20).

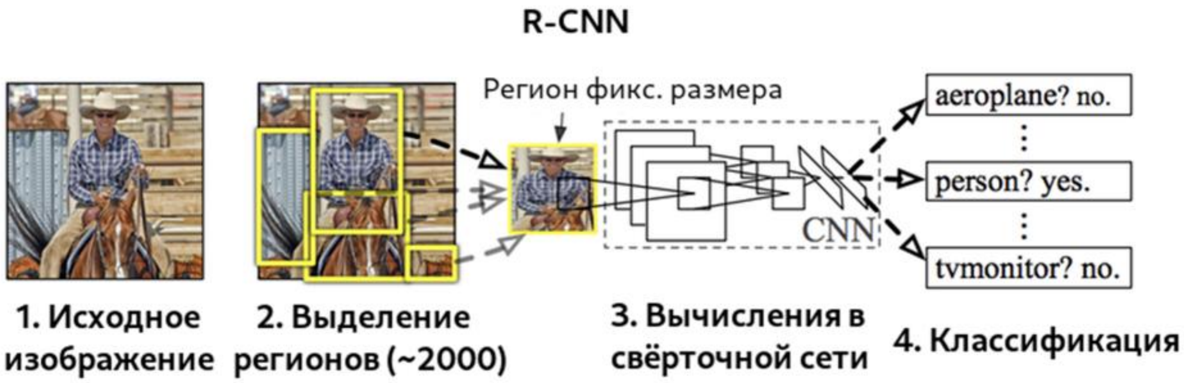
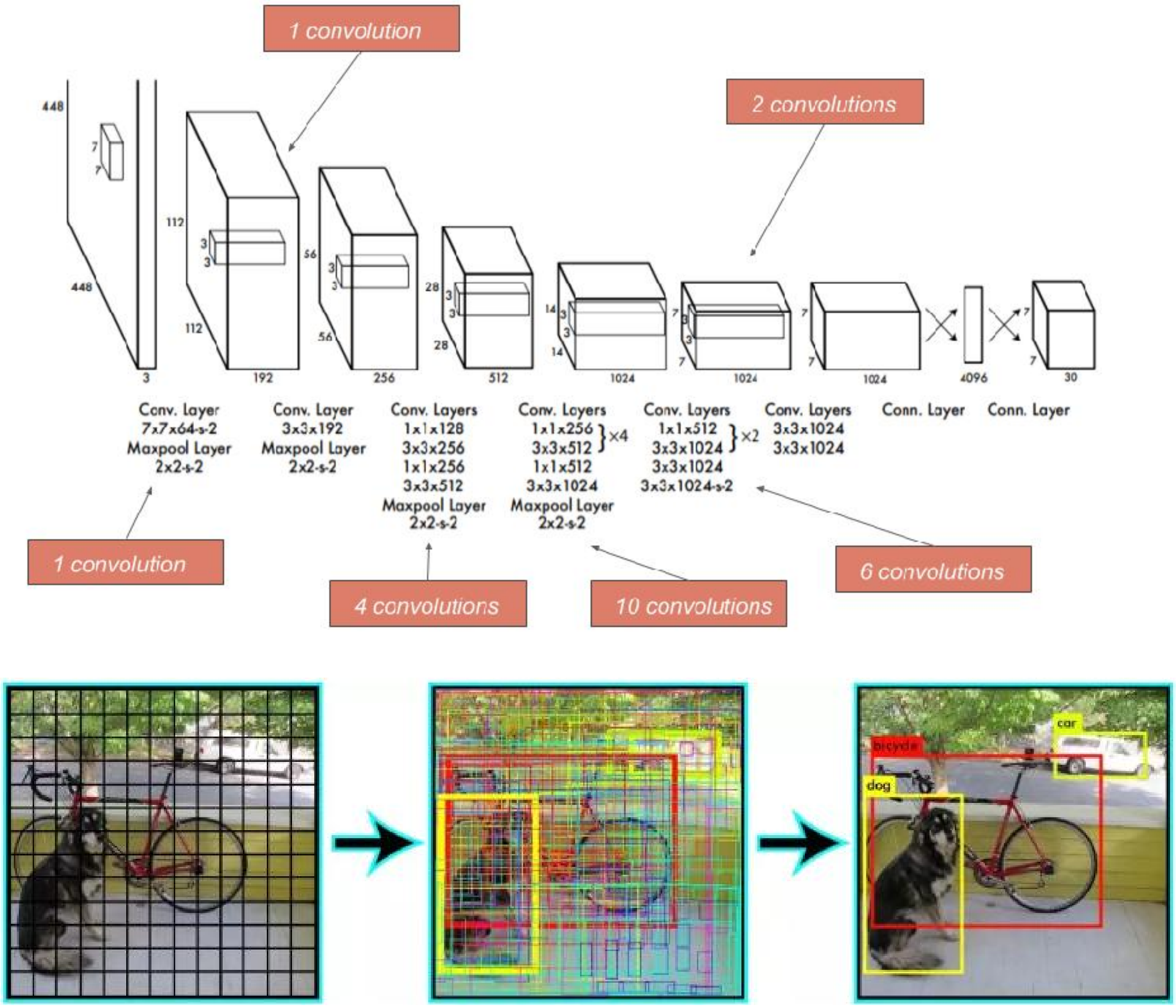
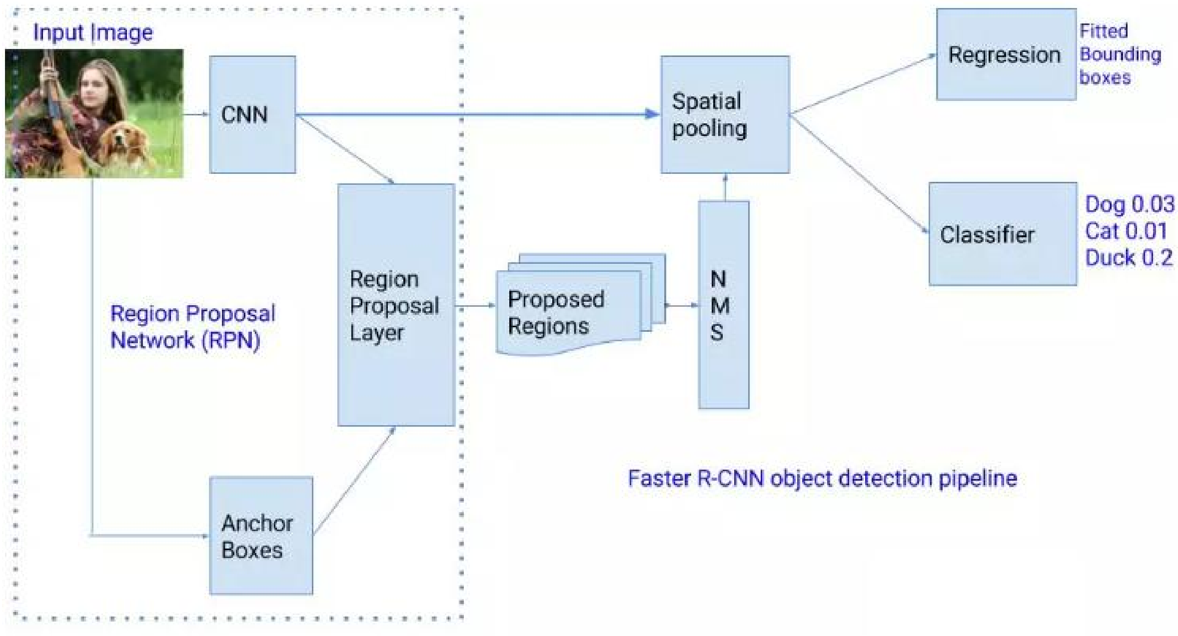


Рисунок 1.20 – Модель R-CNN.

Зовсім недавно було запропоновано підхід до виявлення об’єктів "You Only Look Once" (YOLO), який зменшив обчислювальні складності, пов’язані з R-CNN шляхом формулювання об’єкта. Проблема виявлення як єдина проблема регресії, де координати обмежувальної коробки та ймовірності класу обчислюються в той же час. Незважаючи на те, що YOLO було продемонстровано, що вона надавала значні результати – переваги швидкості нахилу перед R-CNN (наприклад, 45 кадрів в секунду на GPU Nvidia Titan-X) було також показано, що при локалізації похибка YOLO значно вища, ніж новіші варіанти R-CNN, такі як FR-CNN. Регіональна мережа пропозицій (RPN) у програмі FR-CNN передбачає зміщення та конфіденційність для ящикf, що використовують ручні підібрані пріори замість прогнозування обмежувального поля координати безпосередньо (рис. 1.21, рис. 1.22).



а)



б)

Рисунок 1.21 – Моделі YOLO (а), Faster R-CNN (б).

Рисунок 1.22 – Порівняльний аналіз якості і швидкості мереж.

Кожен якір пов’язаний з 4 координатами для поля та 2 бальних значень, які оцінюють ймовірність об’єкту, а не об’єкт запропонованого поля.

Мотивоване вдосконаленням швидшого R-CNN через якір запропонували вдосконалений YOLO (названий YOLOv2), де анкерні коробки використовуються для прогнозування обмежувальних прямокутників (ящиків). Крім того, порівняно з YOLO, YOLOv2 не має повністю пов’язаних шарів у її мережевій архітектурі. Було зроблено процедуру навчання простіше для мережі (k-означає кластеризацію), було застосовано принцип на навчальному наборі обмежувальних коробок автоматично обирати хороші пріоритети і тим самим підвищити точність моделювання.

Щоб зробити мережу швидшою, YOLOv2 використовував нову мережу CNN (тобто, Darknet-19) на відміну від інших рамок, які використовують VGG-16. Більш конкретно, Darknet-19 вимагає 8,52 млрд. операцій з плаваючою точкою, що значно нижче порівняно до VGG-16, що вимагає 30,69 млрд. операцій з плаваючою точкою у кожному передньому проході.

Експериментальні результати для цього підходу продемонстрували, що

YOLOv2 може виконувати виявлення об’єктів зі швидкістю 67 FPS на графічному процесорі Nvidia Titan-X, досягаючи сучасного виявлення.

Хоча YOLOv2 може досягти продуктивності в режимі реального часу на потужному графічному процесорі, вона все ще залишається дуже складною для використання цього підходу для виявлення об’єктів у режимі реального часу у відео на вбудованих обчислювальних пристроях з обмеженою обчислювальною потужністю та обмеженою пам’яттю.

Наприклад, у різних реальних додатках, таких як висновки в режимі реального часу на смартфонах та вбудованому відеоспостереженні. Тому виявлення об’єктів у режимі реального часу у відео на вбудованих пристроях залишається великою проблемою для вирішення.

Вмотивовані вирішити цей виклик, ряд вчених пропонує нову рамку під назвою Fast YOLO, яка прискорює YOLOv2 в режимі реального часу для виявлення заперечень у відео на вбудованих пристроях за допомогою двох ключових стратегій. По-перше, мотивовані перспективними результатами, продемонстрованими Shafiee та ін. пропонується використовувати еволюційну глибоку інтелектуальну структуру для розвитку мережевої архітектури YOLOv2 та створити оптимізовану архітектуру (тут згадується O-YOLOv2), що має значно менші параметри при збереженні високих показників виявлення. По-друге, адаптивний метод висновку вводиться в запропоновану рамку Fast YOLO для скорочення частоти глибокого висновку з O-YOLOv2 на основі аналізу темпу характеристики руху. Поєднання цих двох стратегій приводить до рамки виявлення об’єктів, яка істотно покращує швидкість виявлення на вбудованих пристроях.

Розглянемо базову методологію. Запропонований фреймворк Fast YOLO розділений на два основних компоненти: 1) оптимізована архітектура YOLOv2 та 2) адаптивна до умови руху система (рис. 1.23). Для кожного відеокадру встановлено stac зображення що складається з відеокадру з опорним кадром передається в згортковий шар 1×1. Результатом згорткового шару є карта ймовірності руху, яка потім подається в адаптивний модуль виводу, щоб визначити, чи потрібен глибокий висновок для обчислення оновленої карти ймовірностей класу.

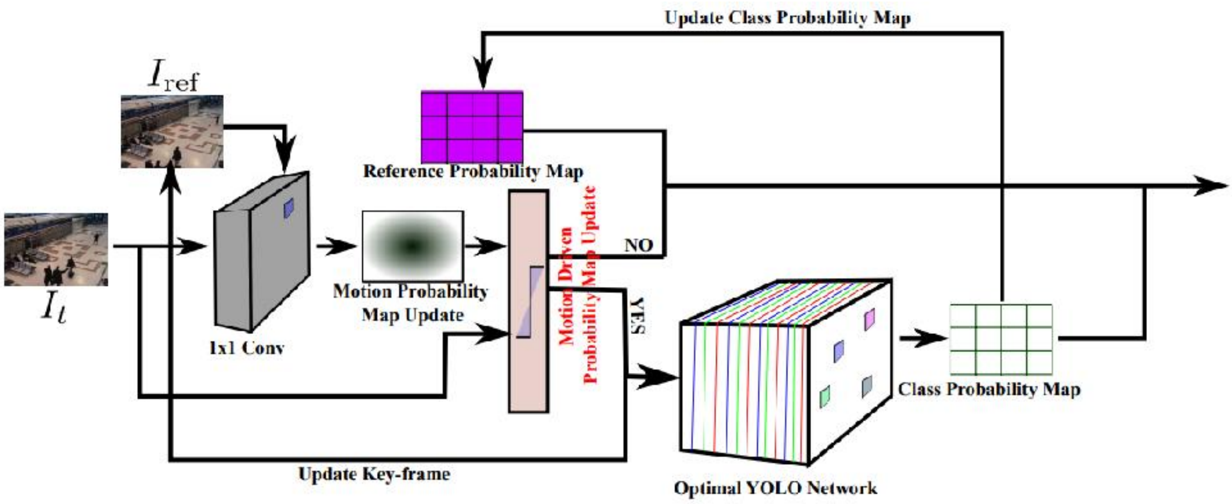


Рисунок 1.23 – Огляд запропонованої рамки Fast YOLO для виявлення об’єктів у відео.

Як було сказано вище, головна мета – ввести рамку для об’єкта виявлення у відео, яке може працювати швидше на вбудованих пристроях, зменшуючи споживання ресурсів, що, в свою чергу, значно зменшує споживання енергії. За допомогою використання цього руху, що адаптується, частота глибоких висновків сильно зменшується і виконується лише при необхідності.

Розглянемо аспекти оптимізації архітектури мережі. Одна з головних проблем у глибоких нейронних мережах, особливо коли використання їх для вбудованих сценаріїв полягає в проектуванні мережевої архітектури. Процес проектування, як правило, виконується експертом з людини, яка досліджує велику кількість конфігурацій мережі, щоб знайти найкращу архітектуру для певного завдання з точки зору точності моделювання та кількості параметрів.

Пошук оптимізованої архітектури зазвичай вирішується як проблема оптимізації гіперпараметрів, але це спосіб вирішення проблеми дуже трудомісткий, і більшість методів або обчислювальне нерозбірливі для великих мережевих архітектур, або призводять до пошуку субоптимальних рішень, недостатні для вбудованого використання. Наприклад, загальним підходом до оптимізації гіперпараметрів є сітка пошуку, де велика кількість різних конфігурацій мережі вивчається, і тоді найкраща конфігурація вибирається як остаточна мережева архітектура за критерієм.

Однак глибока нейронна мережа, призначена для виявлення об’єктів у відео, наприклад, YOLOv2, мають величезну кількість параметрів і як таке не обчислюється обчислювально для пошуку всього простору параметрів, щоб знайти оптимальне рішення. Як такий, а не використовувати гіперпараметр для оптимізаційного підходу до отримання оптимальної мережевої архітектури.

На основі YOLOv2 ми скористаємося стратегіями оптимізації мережі, розробленими спеціально для підвищення ефективності мережі.

В еволюційній глибокій розвідці, архітектурні риси глибокої нейронної мережі моделюються через імовірнісну стратегію моделювання генетичного кодування. Зокрема, імовірнісна "ДНК", вироблена таким чином, кодує ймовірність існування всіх можливих синапсів в архітектурі мережі.

Імовірнісна «ДНК» мережі предків (імітує успадкування) разом із чинниками навколишнього середовища потім використовується щоб синтезувати нову, потомську глибоку нейронну мережу в стохастичній манері (імітуючи природний відбір та випадкову мутацію).

Цей процес кодування та синтезу повторюється покоління за поколінням, виробляючи все більш ефективні глибокі нейронні мережі, як проходить час. Однією з ключових переваг такого підходу є те що, на відміну від підходів до оптимізації гіперпараметрів, потрібно оцінити величезну кількість можливих рішень; тим самим значно зменшується обчислювальна складність пошуку оптимальної архітектури мережі.

Використовуючи еволюційну глибоку розвідку для отримання оптимізованої мережевої архітектури на основі YOLOv2 для вбудованого об’єкта виявлення на відео, ми беремо до уваги той факт, що обчислювальна потужність і наявна пам’ять сильно обмежена на вбудованих пристроях.

Тому ми налаштовуємо фактори навколишнього середовища, використовувані при синтезі мережі, які обробляють таким чином, що кількість параметрів в архітектурі мережі сильно зменшується в кожному поколінні; як і кількість параметрів є основним фактором обчислювальних та пам’ятних вимог глибокої нейронної мережі.

Використання такого підходу дозволяє автоматично знайти оптимізовану мережеву архітектуру на основі YOLOv2 (яку ми будемо називати O-YOLOv2), що містить ~2,8X менших параметрів порівняно з оригінальною мережевою архітектурою YOLOv2.

Тепер, щодо адаптації щодо руху. Для подальшого зменшення споживання енергії процесорного блоку для цілей вбудованого виявлення об’єктів у відео, ми враховуємо те, що не всі зняті відеокадри містять унікальну інформацію і, таким чином, глибокого висновку не потрібно виконувати на всіх кадрах.

Як такий, ми вводимо адаптивний до руху підхід до висновку, щоб визначити, чи потрібен глибокий висновок по конкретному кадру відео.

У мережі O-YOLOv2, яка запроваджена вище, при необхідності, ця методика адаптації до руху може допомогти рамці зменшити попит на обчислювальні ресурси.

Що призводить до значного зниження споживання електроенергії системи, а також до збільшення швидкості обробки.

Процес виведення з адаптацією до руху можна побачити на рис. 1.23. Кожен кадр укладається з опорним кадром Iref для формування стеку зображень. Потім згортковий шар 1×1 виконується на стеку зображень для створення карти ймовірностей руху.

Карта ймовірності руху разом з кадром It передаються в модуль, що адаптується до руху, і визначає, чи є кадр It достатньо унікальний, порівняно з еталонним кадром для отримання глибокого висновку для обчислення нової карти ймовірностей класу.

Якщо модуль визначає це, тоді мережа O-YOLOv2 використовується для обчислення оновленої карти ймовірностей класу.

Потім карта зберігається у вигляді Iref та референтної карти ймовірностей відповідно.

Користуючись лише цією простою, але ефективною процедурою, кадри, які потребують глибокого висновку, обробляються оперативно, що призводить не тільки до зменшення споживання електроенергії, але і до зменшення середнього часу роботи на кадр і може забезпечити більш швидку обробку даних.

Щодо результатів впровадження та експериментів.

Запропоновані рамки Fast-YOLO оцінюються за допомогою двох різних стратегій. Спочатку ми оцінюємо точність моделювання та продуктивність оптимізованої архітектури мережі YOLOv2 (тобто O-YOLOv2) з оригінальною мережевою архітектурою YOLOv2.

Використовуємо набір даних VOC 2007, щоб продемонструвати ефективність роботи мережі, процес оптимізації архітектури. На рис. 1.24 наведено архітектурну та порівняння продуктивності між O-YOLOv2 та оригінальним YOLOv2 на наборі даних VOC.

Можна помітити, що мережева архітектура O-YOLO в 2,8 рази менша порівняно з оригінальним YOLOv2 з лише 2% падінням IOU, що мало б вплив на реальні програми для виявлення об’єктів на основі відео.

По-друге, пропоновані рамки Fast YOLO, O-YOLOv2 та оригінальний YOLOv2 оцінюється за середнім часом роботи на системі Nvidia Jetson TX1. Результати можно побачити на рис. 1.25. Запропонована швидка рамка YOLO може зменшити кількість глибоких висновків в середньому на 68,5%, що призводить до середнього часу роботи 56мс порівняно з 184мс, що досягнуто оригінальним YOLOv2 (прискорення ~3,3X).

Network Architecture	Number of parameters	IOU
YOLOv2	48.2M	67.2%
O-YOLOv2	17.1M	65.10%

Рисунок 1.24 – Архітектурне та продуктивне порівняння між оригінальною мережевою архітектурою YOLOv2 та оптимізованою мережевою архітектурою YOLOv2 (OYOLOv2).

Framework	Frame per Second (FPS)	Inference frequency (%)
YOLOv2	5.40	100
O-YOLOv2	11.80	100
Fast YOLO	17.85	61.87

Рисунок 1.25 – Середні показники часу виконання та частота глибин висновок пропонованих швидких YOLO, O-YOLOv2 та оригіналу YOLOv2 працює на вбудованій системі Nvidia Jetson TX1.

У цьому підрозділі роботи представили Fast YOLO, нову основу для виявлення вбудованого об’єкта в режимі реального часу у відео.

Хоча YOLOv2 розглядається як найсучасніший фреймворк з висновками в режимі реального часу на потужних графічних процесорах, використовувати його на вбудованих пристроях у режимі реального часу неможливо. Тут ми скористаємося еволюційною рамкою глибокого інтелекту для створення оптимізованої мережевої архітектури на основі YOLOv2.

Оптимізована мережева архітектура використовується в межах, орієнтованому на прискорення руху процес виявлення, а також зменшення споживання енергії вбудованого пристрою. Результати експериментів показали, що запропоновані рамки Fast YOLO можуть досягти середнього часу виконання, який в 3,3 рази швидший порівняно з оригінальним

YOLOv2, може зменшити кількість глибоких висновків в середньому на 38,13%, і має мережеву архітектуру, яка на 2,8X більше компактна.

1.3 Сучасний стан і перспективи розвитку методів сегментації

Сегментація зображень посідає значне місце у системах розпізнавання та обробки інформації. Постійний розвиток технологій обробки зображень призвели до виникнення нових підходів до вирішення завдань сегментації зображень і застосування їх при вирішенні багатьох практичних завдань. На теперішній час існує значна кількість методів для проведення сегментації зображень.

В залежності від типу зображення та його використання підходи до сегментації будуть різними. Вибір методу сегментації буде залежати від вимог до швидкості виконання сегментації та необхідності втручання користувача в процес сегментації.

Але як свідчить аналіз багатьох експериментів, найкращі результати показують методи сегментації границь по контрастності на основі використання адаптивних масок. Цей підхід є найбільш перспективним у теперішній час. Від полягає в основі згорткових мереж, і може використовуватися окремо. У чому різниця?

Якщо ми використовуємо нейроні мережі, ми повинні їх обучати. Якщо об’єкти, з якими мі працюємо мають нерегулярний вид, нейронну мережу на такі об’єкти натренувати практично неможливо. У такий ситуації треба переходити к класичному підходу аналізу контрастності.

1.4 Локалізація зображень

Сегментовані зображення самі по собі можуть розглядатися я к результат локалізації. Проте у деяких системах є потреба обробляти зображення об’єкту з урахуванням його контексту (чи околу). У такій ситуації необхідно побудувати окіл сегментованої множини пікселів. Таке завдання можна вирішити декількома. Найпростіше – побудувати ε-окіл границі сегментованого об’єкту. Але на практиці як найчастіше є умова побудувати прямокутну область навколо об’єкту.

Для вирішення цієї задачі будемо використовувати метод головних компонент (МГК), який дозволяє знайти орієнтацію об’єкту у просторі з метою адекватної апроксимації області об’єкту прямокутником (рис. 1.26). Детальніше розглянемо цій аспект у наступному розділі.

	X_1	X_2
1	0.407	0.353
2	0.475	0.355
3	-0.088	-0.045
4	0.394	0.325
5	0.274	0.202
6	0.131	0.258
7	-0.053	-0.031
8	-0.124	-0.128
9	-0.469	-0.344
10	0.088	0.171
11	-0.261	-0.162
12	0.401	0.341
13	-0.376	-0.143
14	-0.251	-0.255
15	-0.325	-0.316
16	0.464	0.248
17	-0.310	-0.207
18	0.307	0.247
19	-0.399	-0.303
20	-0.253	-0.253
21	-0.341	-0.291

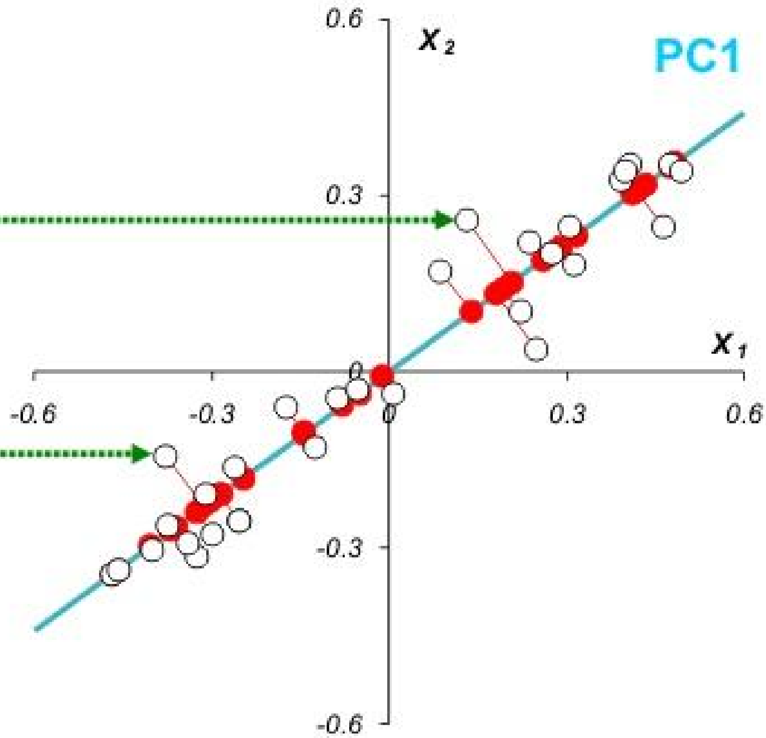


Рисунок 1.26 – Застосування методу головних компонент [20]

Як видно з розгляду рисунка 1.27, використання методу головних компонент дозволяє швидко і адекватно знайти осі для побудови моделі в просторі ознак. А при необхідності ще й оптимізувати чисельність осей, залишив тільки інформативні.

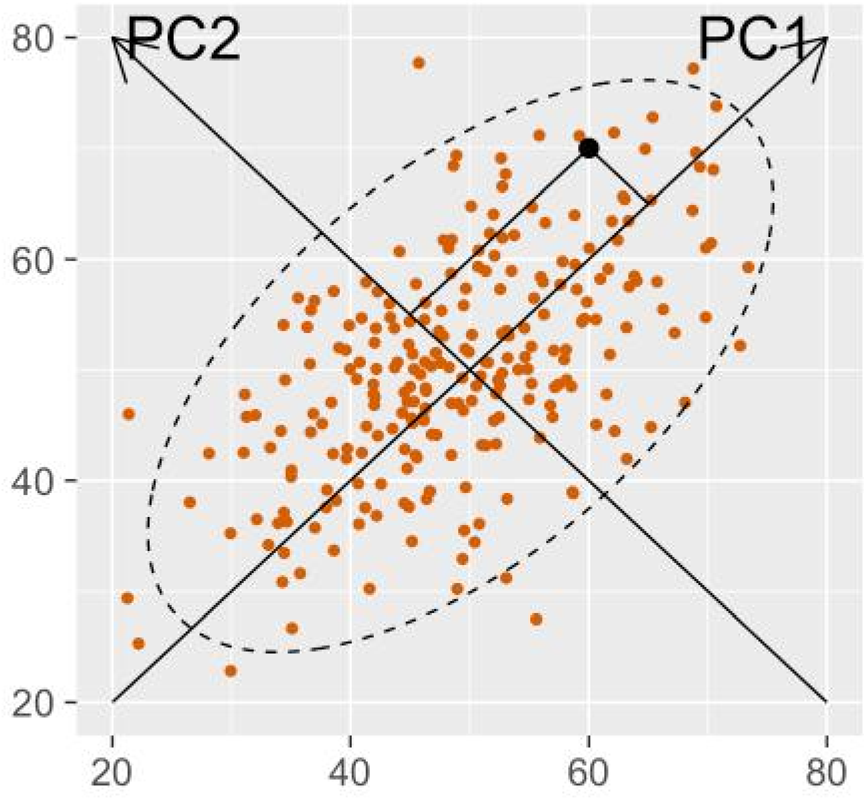


Рисунок 1.27 – Застосування методу головних компонент

2 МОДЕЛІ І АЛГОРИТМИ СЕГМЕНТАЦІЇ І ЛОКАЛІЗАЦІЇ

2.1 Базова модель сегментації зображень

Значна частина завдань обробки інформації й аналізу даних пов’язана із зображеннями. Прикладами можуть служити обробка й аналіз даних дистанційного зондування Землі із супутників, об’ємні зображення об’єктів, отримані за допомогою голографічних пристроїв, результати застосування методів неруйнівних досліджень і контролю в промисловості, розробка «органів зору» роботів і сучасних медичних систем діагностики (рентгенографія, машинна томографія, ендоскопія й т.д.). Для успішного рішення завдань пошуку й ідентифікації об’єктів, визначення різного роду їхніх кількісних характеристик необхідно, щоб первинні зображення характеризувалися високою візуальною якістю, що втрачається через незадовільні умови одержання зображень, недосконалості систем передачі відеоінформації і її відображень, впливи перешкод і т.п.

Виділення контурів (виділення меж) – один із напрямів сегментації зображень, що широко використовується у теорії обробки зображень і комп’ютерного зору, частково з області пошуку об’єктів і виділення об’єктів, та грунтується на алгоритмах, які виділяють точки цифрового зображення, в яких різко змінюється яскравість або є інші види неоднорідності.

Основною метою виявлення різких змін яскравості зображення є фіксація контуру об’єктів зображення. Вони можуть відображати різні припущення про модель формування зображення, зміни в яскравості зображення можуть вказувати на: зміну глибини; зміну орієнтації поверхонь; зміну у властивостях матеріалу; відмінність у висвітленні сцени.

В ідеальному випадку результатом виділення контурів є набір зв’язаних кривих, що позначають межі об’єктів, граней і відбитків на поверхні, а також криві, які відображають зміни положення поверхонь. Таким чином, застосування фільтра виділення меж до зображення може істотно зменшити кількість оброблюваних даних, через те що відфільтрована частина зображення вважається менш значущою, а найбільш важливі структурні властивості зображення зберігаються. Однак, на жаль, не завжди можливо виділити межі на зображенні в картинах реального світу середньої складності. Межі виділені з таких зображень часто мають такі недоліки, як фрагментованість (криві меж не з’єднані між собою), відсутність меж або наявність помилкових, що не відповідають досліджуваному об’єкту, меж.

Межі мають досить важливе значення в багатьох сферах застосування обробки зображень, особливо в системах машинного зору, які аналізують сцени штучних об’єктів при фіксованому освітленні.

Хоча деяка література розглядає виділення ідеальних східчастих меж, кордони на натуральному зображенні зазвичай не такі. На них зазвичай впливає один або кілька таких ефектів: фокусне розмиття через кінцеву глибини різкості зйомки, розмита півтінь від неточкових джерел світла, затінення гладких об’єктів, і тому багато дослідників використовують ступінчастий край, згладжений функцією Гауса (функція помилки), як найпростіше наближення моделі ідеального краю для моделювання розмитих меж у прикладних задачах.

Існує безліч підходів до виділення контурів (меж), але практично всі можна розділити на дві категорії: методи, засновані на пошуку максимумів, і методи, засновані на пошуку нулів. Методи, засновані на пошуку максимумів, виділяють межі за допомогою обчислення «сили краю», зазвичай вирази першої похідної, такого як величина градієнта, і потім пошуку локальних максимумів сили краю, використовуючи передбачуваний напрям межі, зазвичай перпендикуляр до вектора градієнта. Методи, засновані на пошуку нулів, шукають перетин осі абсцис шляхом обчислення другої похідної, зазвичай нулі лапсасіана або нулі нелінійного диференціального виразу. В якості кроку передобробки до виділення меж практично завжди застосовується згладжування зображення, зазвичай фільтром Гауса.

Канні вивчив математичну проблему отримання фільтра, оптимального за критеріями виділення, локалізації та мінімізації кількох відгуків одного краю. Він показав, що шуканий фільтр є сумою чотирьох експонент. Він також показав, що цей фільтр може бути добре наближений першою похідною Гаусіани. Канні ввів поняття Non-Maximum Suppression (придушення не-максимумів), яке означає, що пікселями меж оголошуються пікселі, в яких досягається локальний максимум градієнта в напрямку вектора градієнта. Хоча його робота була проведена на зорі комп'ютерного зору, детектор меж Канні до сих пір є одним з кращих детекторів. Крім особливих локальних випадків важко знайти детектор, який би працював істотно краще, ніж детектор Канні. Послідовність роботи детектора Канні складається з таких етапів.

- Згладжування. Розмиття зображення для видалення шуму.
- Пошук градієнтів. Межі відзначаються там, де градієнт зображення набуває максимальне значення.
- Придушення НЕ-максимумів. Тільки локальні максимуми відзначаються як кордони.
- Подвійна порогова фільтрація. Потенційні межі визначаються порогами.
- Трасування області неоднозначності. Підсумкові межі визначаються шляхом придушення всіх країв, незв'язаних з певними (сильними) межами.

Детектор Канні-Деріше був виведений зі схожого математичного критерію, як і детектор Канні, хоча, відштовхуючись від іншої точки зору, він призвів до набору рекурсивних фільтрів для згладжування зображення замість експоненційних фільтрів і фільтрів Гауса.

Після того, як ми знайшли силу межі (зазвичай - величину градієнта), наступний крок - застосувати поріг, щоб вирішити, знаходиться чи ні межа в даній точці зображення. Чим менший поріг, тим більше пікселів буде перебувати на межі, але тим більш сприйнятливим до завад стане результат, виділяючи зайві дані зображення. Навпаки, високий поріг може пропустити слабкі краї або отримати межу фрагментами.

Якщо поріг застосовується просто до зображення величини градієнта, отримані межі будуть товстими і буде потрібно деяка постобробка, що робить край тонким і точним. Якщо ж виділити межі за допомогою Non-Maximum Suppression, кордони будуть тонкими за визначенням і їх можна буде поєднати в полігони процедурою з’єднання країв (простежування межі). На дискретній сітці етап придушення немаксимумів може бути реалізований за допомогою оцінки напрямку градієнта, використовуючи перші похідні, округлення направлення на значення з кроком 45 градусів і, нарешті, порівнянні значень градієнта в отриманому напрямку градієнту.

Традиційним підходом до вирішення проблеми знаходження відповідного порогу є пороги «з запізненням». Метод використовує декілька порогів. Ми використовуємо верхній поріг, щоб знайти точку початку межі. Після того, як ми отримали стартову точку, ми відстежуємо кордон, точка за точкою, поки значення сили краю вище нижнього порогу. Цей алгоритм має на увазі, що межі - це швидше за все безперервні криві, і дозволяє нам простежувати слабкі ділянки меж без допущення того, що всі значні точки на зображенні будуть позначені як край. Однак, у нас все ще є проблема вибору відповідних значень порогів для цього методу, так як оптимальні параметри можуть змінюватися від зображення до зображення.

Уточнення контуру (межі) - процес, який робить контури тонкими, видаляючи небажані хибні точки, які з’являються на межі. Ця техніка застосовується вже після того, як зображення було згладжено (використовуючи фільтр Гауса), був застосований оператор меж (як один з описаних вище) для обчислення сили краю і після того, як межі були очищені, використовуючи відповідні пороги. Цей метод видаляє всі небажані точки і при акуратному застосуванні видає контури товщиною в один піксель.

Плюси: різкі й тонкі межі дозволяють підвищити ефективність розпізнавання об’єктів; при використанні трансформації Хафа для виявлення прямих або еліпсів, тонкі межі дають значно кращі результати; якщо межа являє собою межа деякої області, тонкі межі дозволяють обчислити такі параметри, як периметр, без якоїсь складної арифметики.

Вказані підходи зазвичай використовують для реалізації завдання по виділення границь об’єктів комп’ютерних зображень, як правило, наступним чином:

1) Переведення зображення із моделі кольорів RGB в YUV;

2) Видалення можливих завад зображення шляхом фільтрації;

3) Згладження зображення з використанням фільтрів Гауса;

4) Просторове диференціювання зображення;

5) Пошук локальних максимумів та мінімумів;

6) Зв’язування контурів (пошук границь об’єктів від знайдених максимумів та мінімумів);

7) Кластеризація зображення;

8) Накладання отриманих границь об’єктів на початкове зображення;

9) Виведення сегментованого зображення.

Відмінність такої моделі у тому, що введено процедуру фільтрації зображення від можливих завад та введено процедуру кластеризації об’єктів зображення з метою спрощення подальшого використання сегментованого зображення у системах розпізнавання, а також вилучені процедури із подавленням НЕ-максимумів.

2.2 Виділення границь за допомогою операторів

Одним із підходів до пошуку однорідних областей шляхом виділення контурів є використання операторів. Таким є оператори Собеля, Преввита (рис. 1.1) які використовується в області обробки зображень для виділення контурів об’єктів на зображенні. Ці оператори досить часто застосовують в алгоритмах виділення меж. По суті, це дискретний диференціальний оператор, який обчислює наближене значення градієнта яскравості зображення. Результатом застосування оператора Собеля в кожній точці зображення є або вектор градієнта яскравості в цій точці, або його норма. Оператор Собеля заснований на згортку зображення невеликими сепарабельним цілочисельними фільтрами у вертикальному та горизонтальному напрямках, тому його відносно легко обчислювати. З іншого боку, використовувана ним апроксимація градієнта досить груба, особливо це позначається на високочастотних коливань зображення.

Якщо простіше, то оператор обчислює градієнт яскравості зображення в кожній точці. Так знаходиться напрям найбільшого збільшення яскравості і величина її зміни в цьому напрямі. Результат показує, наскільки «різко» або «плавно» змінюється яскравість зображення в кожній точці, а отже, ймовірність знаходження точки на межі, а також орієнтацію межі. На практиці, обчислення величини зміни яскравості (імовірності належності до грані) надійніше і простіше в інтерпретації, ніж розрахунок напрямку.

Математично, градієнт функції двох змінних для кожної точки зображення (якої і є функція яскравості) - двовимірний вектор, компонентами якого є похідні яскравості зображення по горизонталі й вертикалі. У кожній точці зображення градієнтний вектор орієнтований у напрямку найбільшого збільшення яскравості, а його довжина відповідає величині зміни яскравості. Це означає, що результатом оператора Собеля в точці області постійної яскравості буде нульовий вектор, а в точці, що лежить на кордоні областей різної яскравості - вектор, що перетинає межу у напрямку збільшення яскравості.

Оскільки функція яскравості відома тільки в дискретних точках, ми не можемо визначити похідні до тих пір, поки ми покладемо яскравість безперервною функцією, яка проходить через ці точки. З цієї додаткової передумовою похідну безперервної функції яскравості можна обчислити як від функції, з якої взято виміри - точки зображення. Виявляється, що похідні в будь-якій окремій точці є функції яскравості від всіх точок зображення.

Однак наближення їх похідних можна визначити з більшою чи меншою мірою точності.

Оператор Собеля представляє собою більш наочне наближення градієнта зображення, але він досить якісний для практичного застосування у багатьох завданнях. Точніше, оператор використовує значення інтенсивності тільки в околиці 3 × 3 кожного пікселя для отримання наближення відповідного градієнта зображення, і використовує тільки цілочисельні значення вагових коефіцієнтів яскравості для оцінки градієнту.

Як випливає з визначення, оператор Собеля можна реалізувати простими технічними та програмними засобами: для наближення вектор-градієнта потрібні тільки вісім пікселів навколо точки зображення і цілочисельна арифметика. Більше того, обидва дискретних фільтра, описаних вище, можна розділити на дві групи. Роздільність цих обчислень може призвести до зменшення арифметичних дій з кожним пікселів.

Результат застосування оператора Собеля є двовимірна карта градієнта для кожної точки. Її можна обробити і показати як картинку, на якій ділянки з великою величиною градієнта (в основному, грані) будуть видні як білі лінії. Оператор Собеля згладжує паразитні ефекти на зображенні, що викликаються чисто центрально-диференціальним оператором, але не має повної обертальної симетрії.

Оператор Превітт – метод виділення меж в обробці зображень, який обчислює максимальний відгук на множині ядер згортки для знаходження локальної орієнтації межі в кожному пікселі. Він був створений доктором Джудіт Превітт (Judith Prewitt) для виявлення меж медичних зображень.

Для цієї операції використовуються різні ядра. З одного ядра можна отримати вісім, переставляючи коефіцієнти по колу. Кожен результат буде чутливий до напрямку межі від 0 до 315 з кроком в 45, де 0 відповідає вертикальній межі. Максимальна відповідь кожного пікселя є значення відповідного пікселя у вихідному зображенні. Значення його лежать між 1 і 8, в залежності від номера ядра, що дав найбільший результат.

Цей метод виділення меж також називається підстановкою шаблонів меж (англ. edge template matching), оскільки зображенню зіставляється з набором шаблонів, і кожен представляє деяку орієнтацію межі. Величина і орієнтація межі в пікселі тоді визначається шаблоном, який краще за всіх відповідає локальній околиці пікселя.

Детектор меж Превітт є підходящим способом для оцінки величини та орієнтації межі. У той час як детектор з диференціальним градієнтом потребує трудомісткого обчислення оцінки орієнтації за величинами у вертикальному та горизонтальному напрямках, детектор меж Превітт дає напрямок прямо з ядра з максимальним результатом. Набір ядер обмежений 8 можливими напрямами, проте досвід показує, що більшість прямих оцінок орієнтації теж не дуже точні. З іншого боку, набір ядер потребує 8 згорток для кожного пікселя, тоді як набір ядер градієнтного методу вимагає тільки 2: чутливих по вертикалі і по горизонталі. Результат для зображення потужності меж дуже схожий в обох методів, якщо в них використовуються ті ж ядра згортки.

Оператор Собеля використовує два ядра 3 × 3, згортаючи початкове зображення для обчислення наближених значень похідних: одне по горизонталі і одне по вертикалі.

Застосування оператора Превітт дає кращі результати тоді, коли на зображенні необхідно виділити межу з переважаючим напрямком орієнтації. Розглянуті оператори будуть використані при створені програми для виділення контурів комп’ютерних зображень.

З метою поліпшення якості сегментації за рахунок підвищення чутливості до не контрастних перепадів яскравості будемо використовувати лапласіан гауссіана. Лапласіан зображення в пікселі має вигляд:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}; \tag{2.1}$$

для найбільш поширених згорток 3x3 матриця лапласіану представлена на рисунку 2.1.

На основі використання цього оператору [1] вводиться до розгляду оператор лапласіан гауссіана:

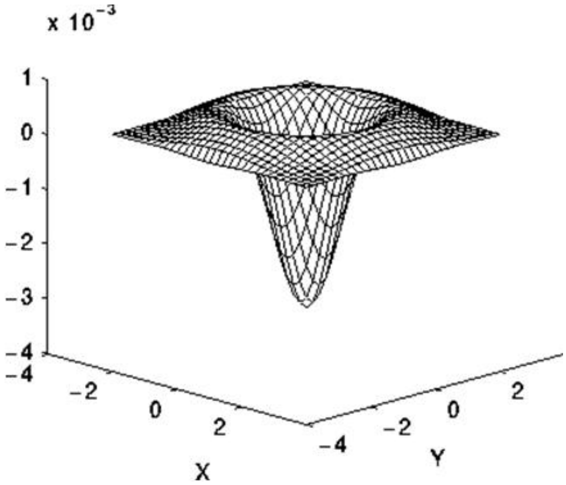
$$\mathrm{LoG}(x, y) = \frac{-1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right] + e^{-\frac{x^2 + y^2}{2\sigma^2}}; \qquad (2.2)$$

геометрична інтерпретація якого наведено на рисунку 2.2.

0	–1	0
–1	4	–1
0	–1	0

–1	–1	–1
–1	8	–1
–1	–1	–1

Рисунок 2.1 – Найбільш поширені дискретні апроксимації оператору Лапласа



а)

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

б)

Рисунок 2.2 – Геометрична інтерпретація оператору LoG і його матриця перетворень, де $\sigma = 1.4$

Обраний оператор дозволяє дуже якісно виділяти граничні профілі зображень, у тому слабо контрастні профілі, завдяки підвищенню різкості границь таким чином, як це наведено на рис 2.3. Приклади застосування оператору наведено на рисунку 2.4.

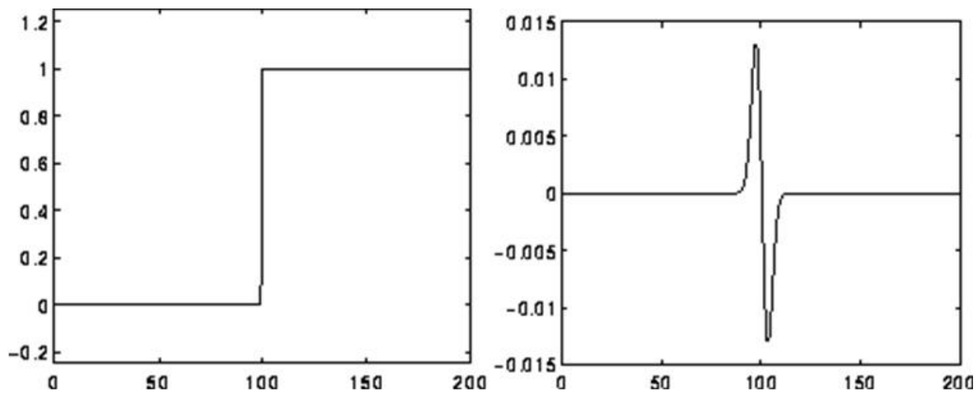


Рисунок 2.3 – Формування граничного переходу оператором LoG

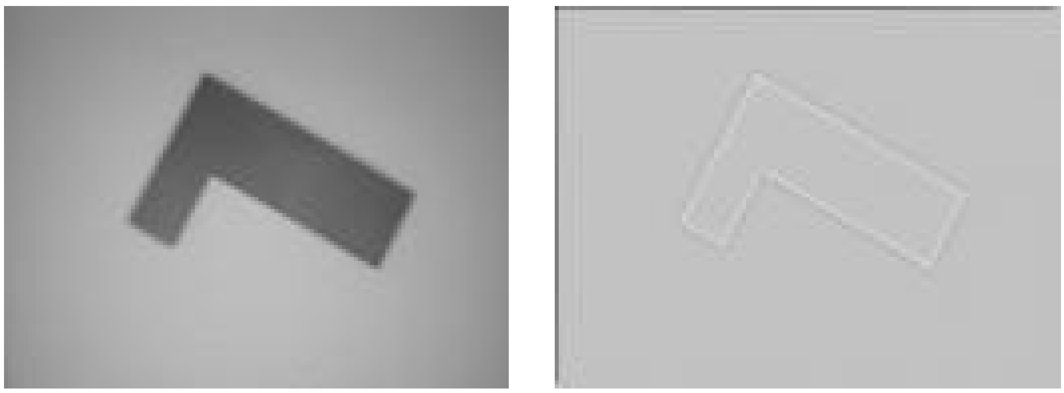


Рисунок 2.4 – Застосування оператора LoG

2.3 Розробка алгоритму сегментації зображення

Знаходження границі зображення (рис. 2.5) починається з находження граничних пікселів. Але для високої ефективності контурної сегментації (для мінімізації рівнів часткової і хибної сегментації) до її початка експерт будує границі зображень (рис. 2.6) з метою настроювання параметрів моделі (порогів) таким чином, щоб мінімізувати відхилення сегментованої границі і талону, який побудував експерт.

Таким чином, маючи еталон (рис. 2.6) ми спроможні ітераційне варіювати пороги і оптимально настроюватися на еталон (рис. 2.7). Далі для сегментації використовують ці оптимальні значення порогів, які дають мінімальні рівні часткової і хибної сегментації (рис. 2.8).

Після отримання первинних результатів сегментації з використанням оператору лапласіан гауссіана другий крок – це застосування методу водорозділів для отримання замкнутих контурів, які відповідають об’єктам на фотографії. Цей метод доцільно застосовувати тому, що об’єкти розташовані на однорідному фоні.

Після цього застосовуємо метод хвиль для отримання зв’язних областей, які відповідають об’єктам.

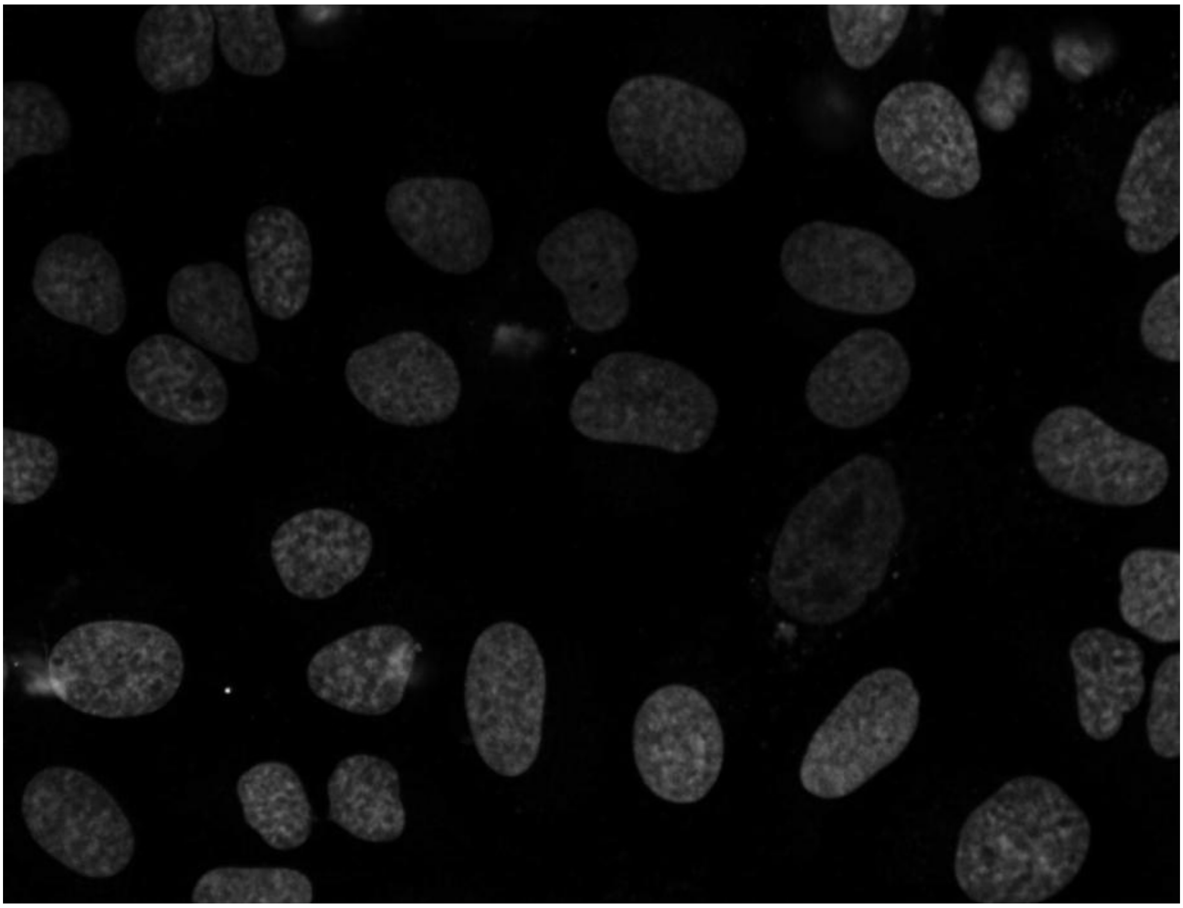


Рисунок 2.5 – Вихідне зображення

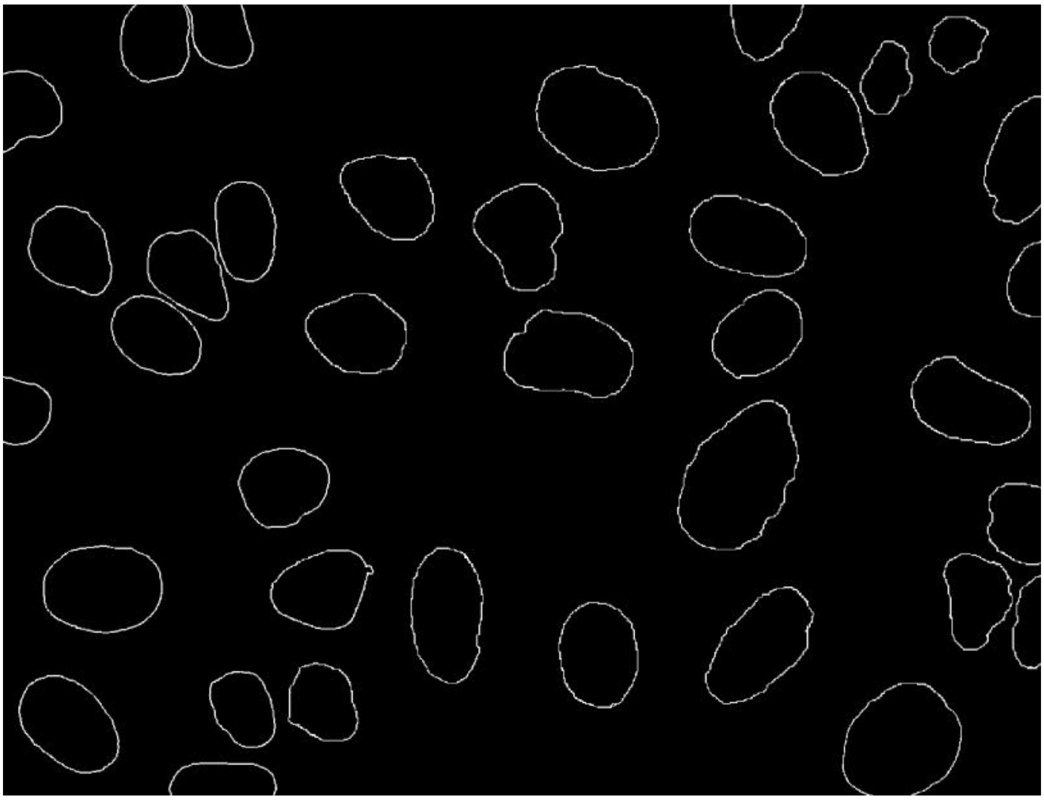


Рисунок 2.6 – Експертне оцінювання границь зображень

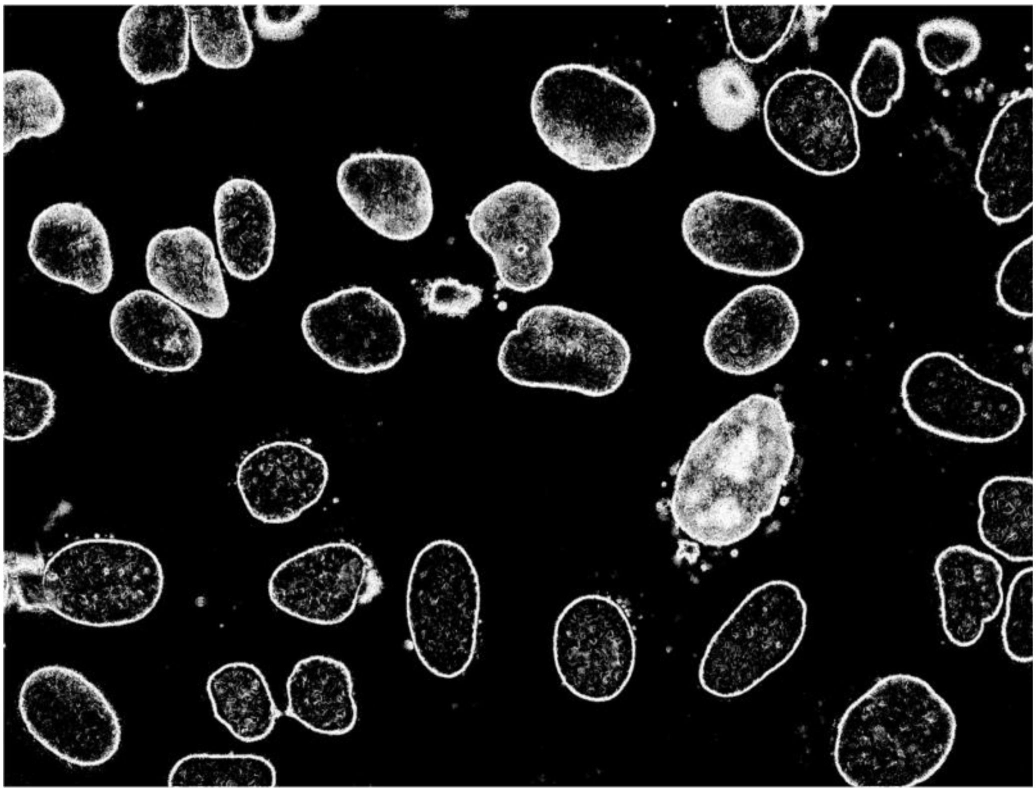
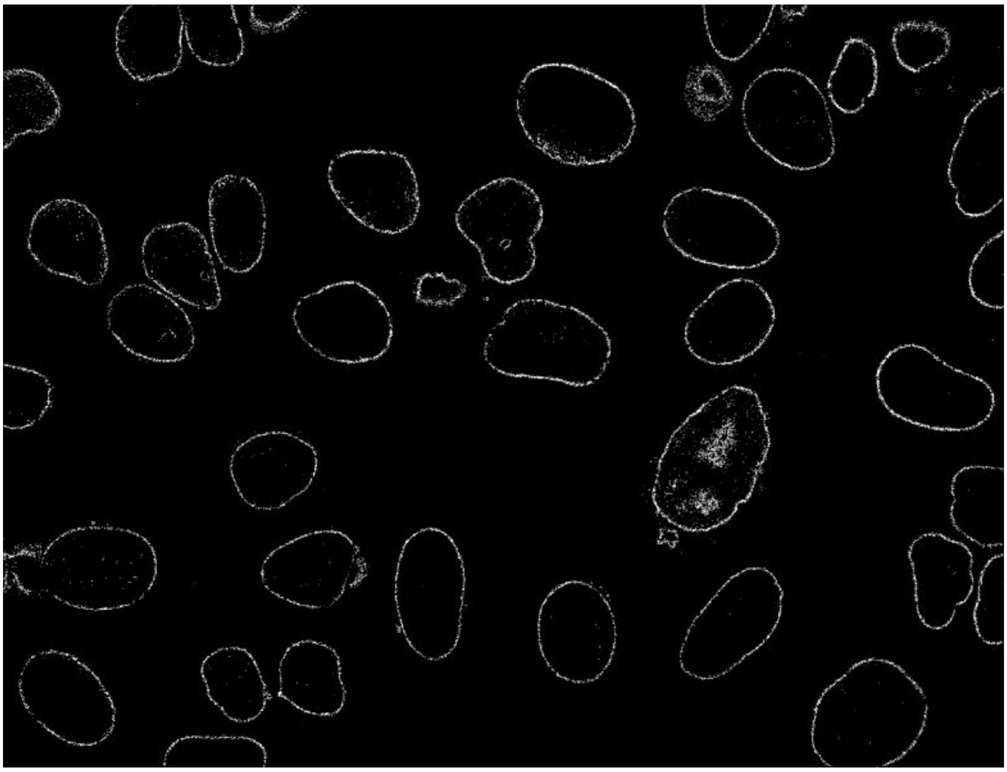


Рисунок 2.7 – Ілюстрація процесу настроювання порогів



9999999999999999999999999999

Рисунок 2.8 – Сегментація границі

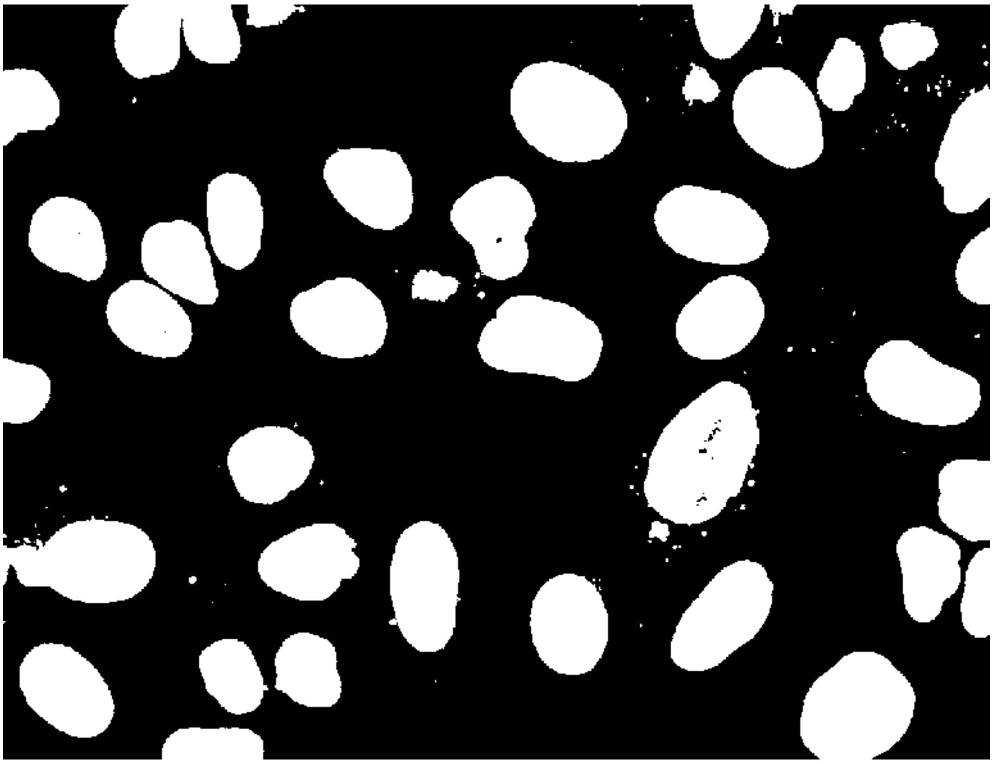


Рисунок 2.9 – Сегментація границі з наявністю об’єктів і хибних фрагментів вихідного зображення

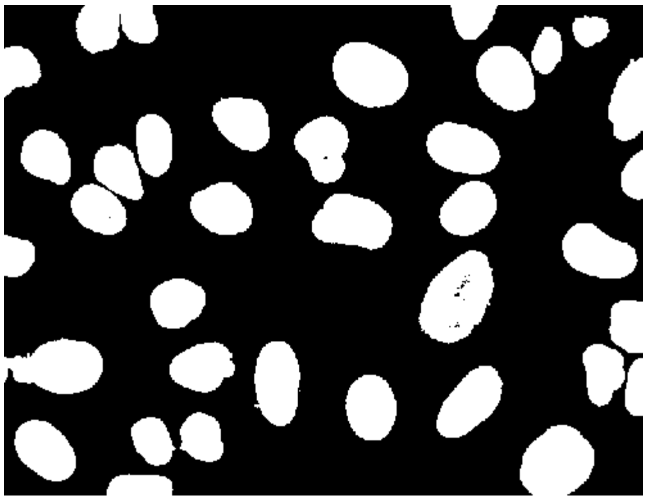


Рисунок 2.10 – Зображення після фільтрації хибних компонент

Наші зображення завжди мають завади і шуми. Наявність яких приводить до хибної сегментації фрагментів вихідного зображення (рис. 2.9). Тому, після застосування методу хвиль застосовуємо геометричну фільтрацію з метою усунення усіх зв’язних компонент, які не є об’єктами. Для такої фільтрації достатньо використовувати параметр площини зображення (рис. 2.10). Наступний крок – застосування морфологічної фільтрації (операція дилатації – зростання світлих областей; ядро 3х3) для видалення «дірок» з нутрощі об’єктів (рис. 2.11 – рис. 2.13).

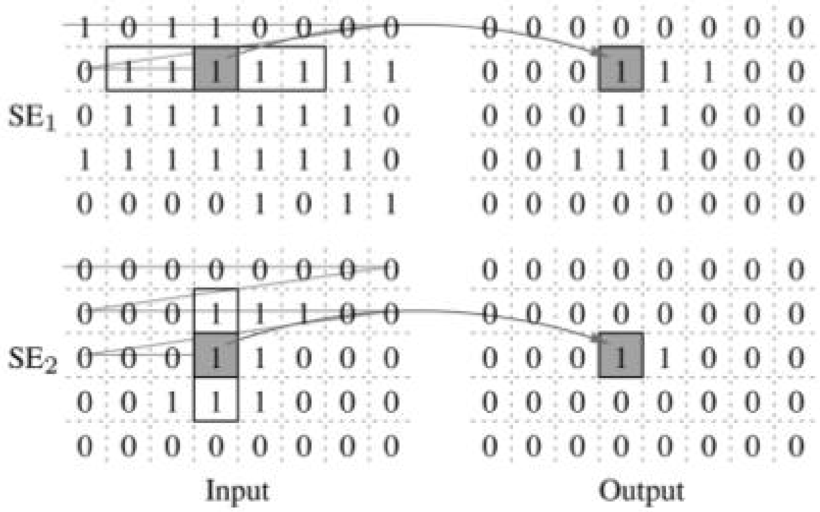
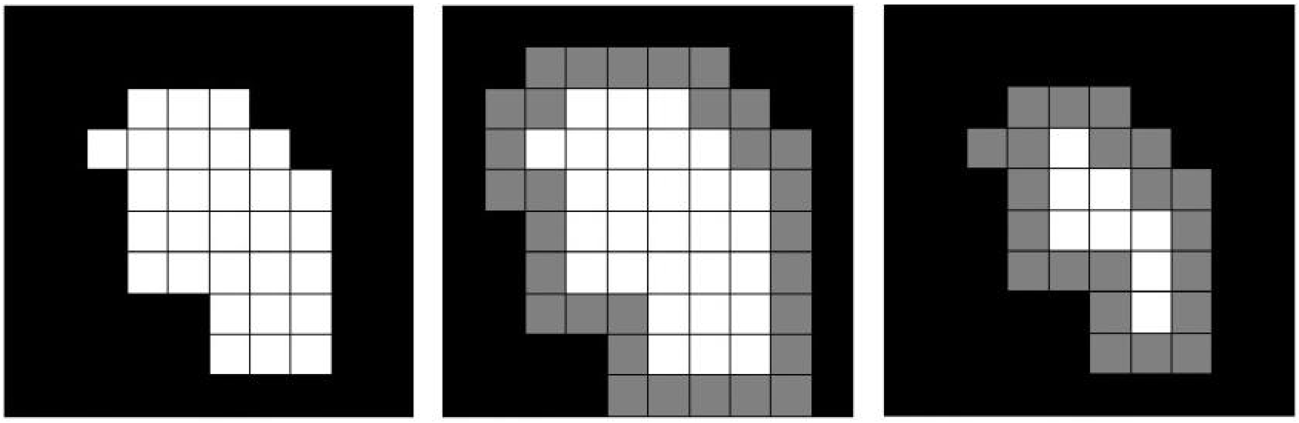


Рисунок 2.11 – Морфологічна фільтрація з усуненням дірок (фільтр Open)



а) б) в)

Рисунок 2.12 – Вихідне зображення (а), дилатація (б) і ерозія (в) зображення

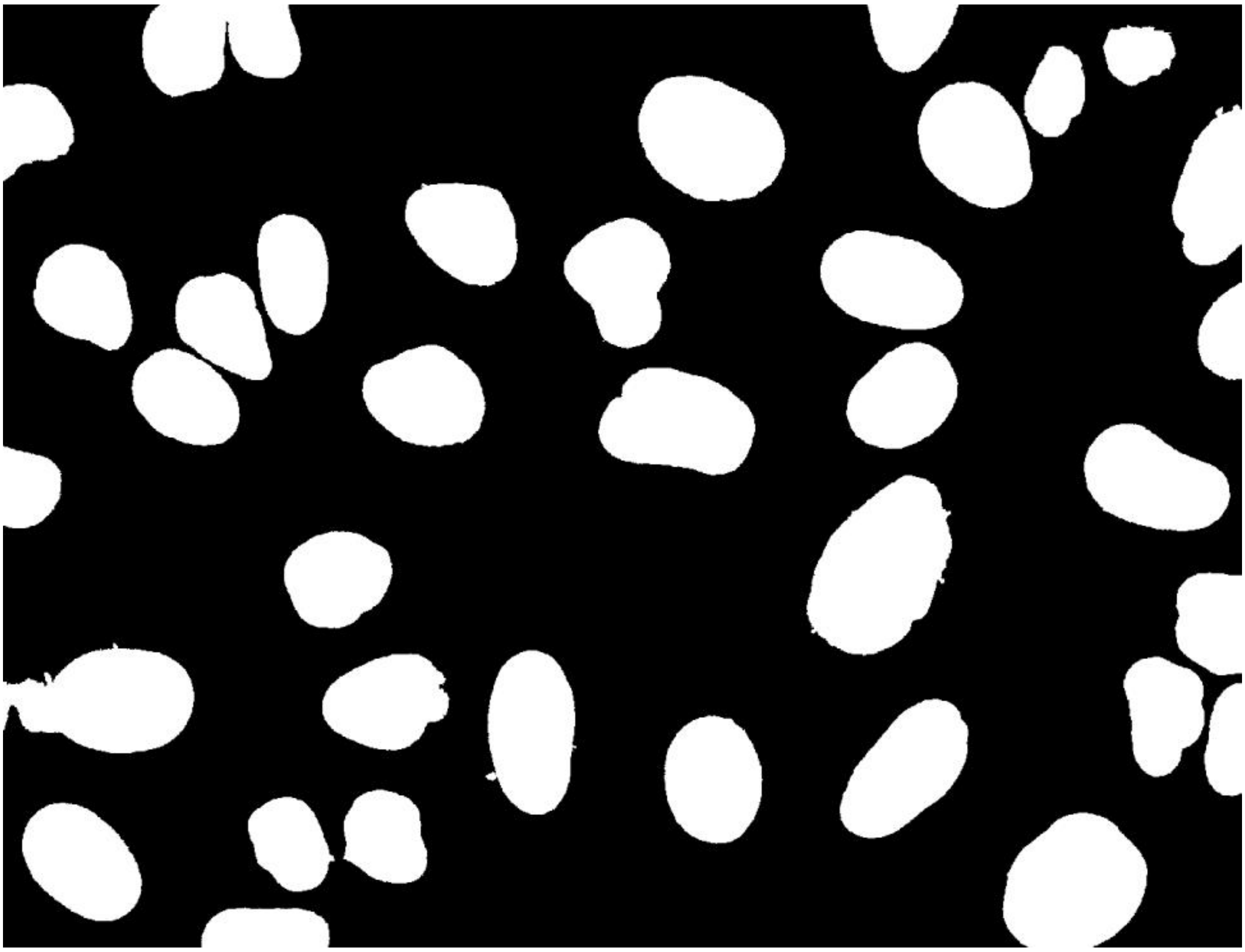
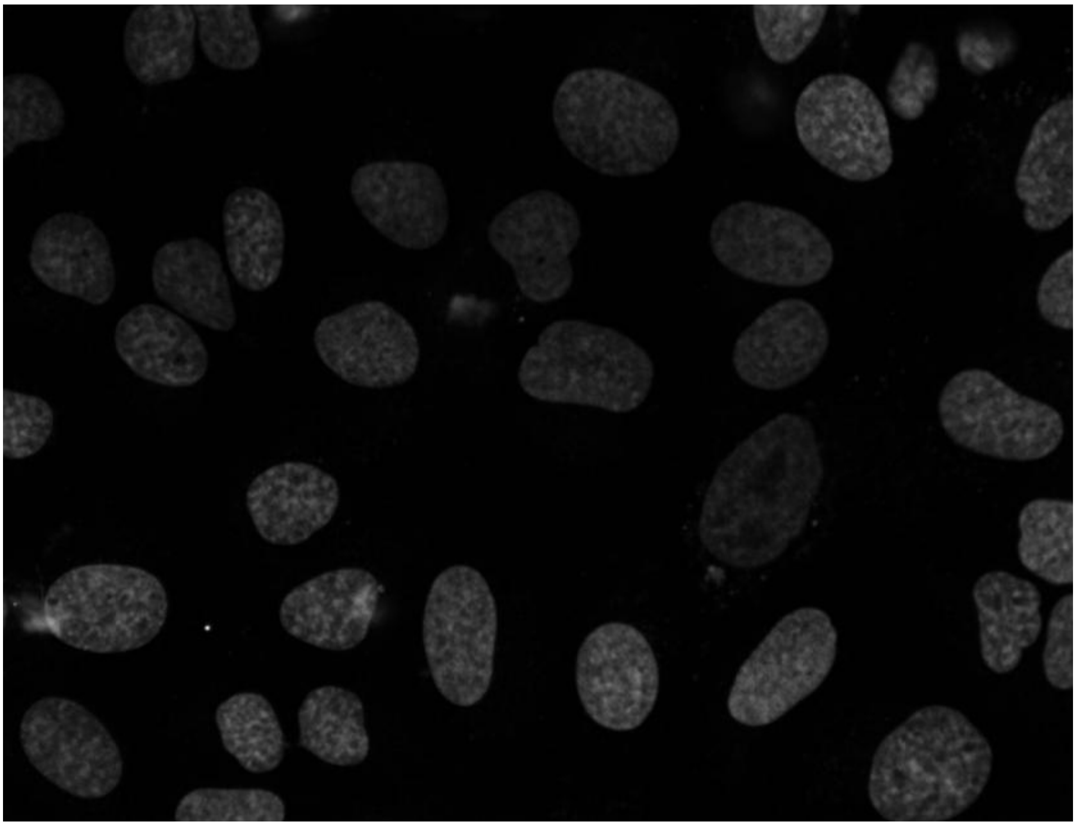
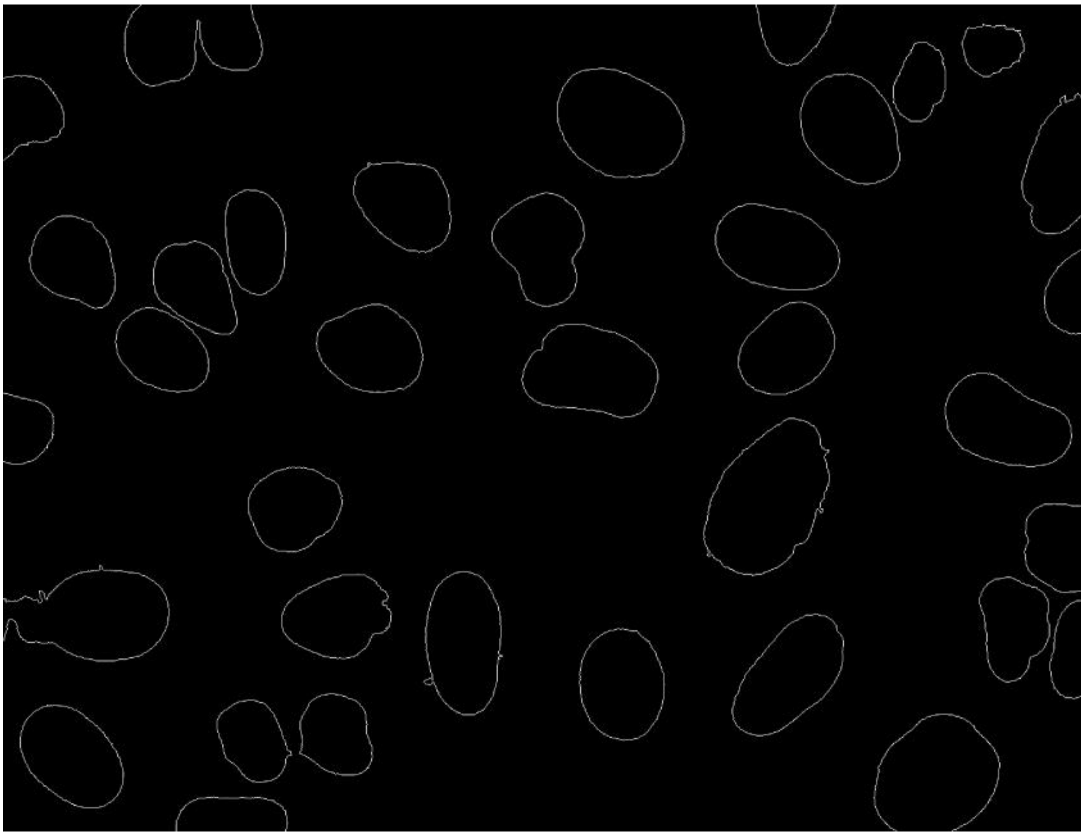


Рисунок 2.13 – Зображення після морфологічної фільтрації і видалення дірок



а)



б)

Рисунок 2.14 – Вихідне зображення (а) і його контурна сегментація (б)

Аналіз результатів обчислювального експерименту дозволяє зробити висновок про те, що запропонована модель (алгоритм) адаптивної сегментації є обчислювальне ефективною і здатною адекватно сегментувати контур об’єкта нерегулярного вигляду в умовах існування завад, а також в умовах істотних варіацій яскравості об’єктів і фону.

2.4 Локалізація з використанням методу головних компонент

Оскільки двовимірний простір ознак являє типовий випадок, розглянемо спочатку цей метод для простору R^2 з координатами (r_1, r_2) з метою отримання формул, використання яких дозволяє підвищити його обчислювальну ефективність.

Для отримання більш компактного покриття значень ознак $\mathbf{X}=\{\mathbf{x}_i = (x_{1i}, x_{2i})\}_{i=1,\dots,n}$, заданих у вихідній системі координат (r_1, r_2), яка визначається базисними векторами $Y = (\mathbf{y}_1, \mathbf{y}_2)$, шукається новий базис $Z = (\mathbf{z}_1, \mathbf{z}_2)$, в якому аппроксимирующий паралелепіпед матиме меншу площу. При $k > 2$ ортогональні вектори, що визначають новий базис, знаходяться послідовно, за критерієм максимального розсіювання вздовж осей; в двовимірному ж випадку нова система координат в базисі Z з точністю до орієнтації осей визначається першої головній компонентою $\mathbf{z}_1 = (z_{11}, z_{12})$, яка максимізує вибіркову дисперсію проекції даних (рис. 2.15), а друга компонента $\mathbf{z}_2 = (z_{21}, z_{22})$, з точністю до орієнтації, найбільш просто визначається з умови ортогональності, наприклад, $z_{21} = -z_{12}$, $z_{22} = z_{11}$.

Перехід до базису $Z = (\mathbf{z}_1, \mathbf{z}_2)$ підвищує адекватність геометричній моделі безлічі $\mathbf{X}$ за рахунок виключення з розгляду областей, значення ознак в яких не відповідають стандартам.

Застосування методу головних компонент передбачає попереднє центрування даних. Для цього знаходимо незміщене положення початку системи координат $\mathbf{O}' = (\bar{x}_1, \bar{x}_2)$:

$$\bar{x}_1 = \frac{1}{n}\sum_{i=1}^{n} x_{1i}\,,\ \bar{x}_2 = \frac{1}{n}\sum_{i=1}^{n} x_{2i}\,, \qquad (2.3)$$

і центруємо дані щодо початку координат $\mathbf{O}' = (\bar{x}_1, \bar{x}_2)$

$$\mathbf{x}'_i = (x_{1i} - \bar{x}_1, x_{2i} - \bar{x}_2)\,,\ (i = 1,2,...,n). \qquad (2.4)$$

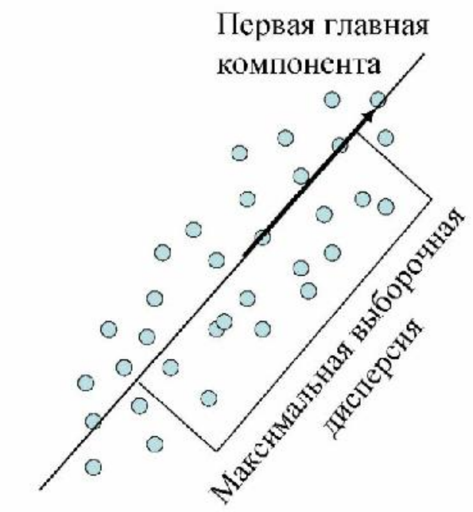


Рисунок 2.15 – Знаходження головних компонент (http://ru.wikipedia.org)

Для спрощення індексації покладемо, що координати точок, складляющих безліч $\mathbf{X}$, задані саме в цій системі координат.

Знаходження компонент перетворення до головних осей $(\mathbf{z}_1, \mathbf{z}_2)$ зводиться до знаходження власних значень $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ і власних векторів $V = (\mathbf{v}_1, \mathbf{v}_2)$ коваріаційної матриці ознак

$$\mathbf{C} = \begin{bmatrix} a & c \\ c & b \end{bmatrix}, \qquad (2.5)$$

де діагональні елементи визначають дисперсії ознак

$$a = \sigma_{11} = \frac{1}{n}\sum_{i=1}^{n}(x_{1i} - \overline{x}_1)^2\,,\; b = \sigma_{22} = \frac{1}{n}\sum_{i=1}^{n}(x_{2i} - \overline{x}_2)^2\,, \tag{2.6}$$

а інші елементи утворюють їх взаємні ковариації

$$c = \sigma_{12} = \sigma_{21} = \frac{1}{n}\sum_{i=1}^{n}(x_{1i} - \overline{x}_1)\cdot(x_{2i} - \overline{x}_2)\,. \tag{2.7}$$

Оскільки для зосереджених даних значення середніх $\overline{x}_1, \overline{x}_2$ дорівнюють нулю, останні вираження приймуть вид

$$\sigma_{11} = \frac{1}{n}\sum_{i=1}^{n}(x_{1i})^2\,,\; \sigma_{22} = \frac{1}{n}\sum_{i=1}^{n}(x_{2i})^2\,, \tag{2.8}$$

$$\sigma_{12} = \sigma_{21} = \frac{1}{n}\sum_{i=1}^{n}x_{1i}\cdot x_{2i}\,. \tag{2.9}$$

Якщо **E** - одинична матриця, то власні значення $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)$ та вектори $V = (\mathbf{v}_1, \mathbf{v}_2)$ матриці **C** знаходяться з рівняння

$$\left(\mathbf{C} - \lambda\cdot\mathbf{E}\right)\cdot\mathbf{v} = 0\,, \tag{2.10}$$

яке представляє собою однорідну систему лінійних рівнянь, нетривіальні рішення якої є шуканими власними векторами матриці.

В явному вигляді рівняння (2.10) представляється в матричному вигляді так

$$\left(\begin{bmatrix} a & c \\ c & b \end{bmatrix} - \lambda\cdot\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)\cdot\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} a-\lambda & c \\ c & b-\lambda \end{bmatrix}\cdot\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0\,, \tag{2.11}$$

де v_1, v_2 позначають компоненти власного вектора $\mathbf{v}_\xi$, $\xi \in \{1,2\}$, або системою лінійних рівнянь

$$\begin{cases} (a-\lambda)\cdot v_1 + \qquad c\cdot v_2 = 0, \\ \qquad c\cdot v_1 + (b-\lambda)\cdot v_2 = 0, \end{cases} \tag{2.12}$$

яке визначає характеристичне рівняння для власних значень виду

$$\lambda^2 - (a+b)\cdot\lambda + (a\cdot b - c^2) = 0,$$

з дискримінантом

$$D = (a-b)^2 + 4c^2 > 0,$$

коріння якого мають вигляд

$$\begin{cases} \lambda_1 = \dfrac{(a+b)+\sqrt{D}}{2}, \\ \lambda_2 = \dfrac{(a+b)-\sqrt{D}}{2}. \end{cases} \tag{2.13}$$

Використання цих значень в системі (2.12), визначає два підпростору власних векторів, що відповідають значенням λ_1 і λ_2, в кожному з яких вектори лінійно залежні, а значить для побудови нового базису $Z = (\mathbf{z}_1, \mathbf{z}_2)$ може вибиратися будь-який з них. Для отримання ортонормованій системи виберемо одиничні вектори. При цьому перший власний вектор $\mathbf{v}_1$ матриці $\mathbf{C}$ відповідає максимальному власному значенню λ_1, а другий – значенню λ_2, або ортогональним йому одиничним векторам.

Таким чином, підставляючи значення λ_1 з (2.12) і поповнюючи систему умовою нормування:

$$\sqrt{v_{11}^2 + v_{12}^2} = 1,$$

отримаємо, що в старих координатах нормований вектор головної компоненти, що визначає вісь z_1, виражається в такий спосіб:

$$\begin{cases} z_{11} = \sqrt{\dfrac{c^2}{c^2 + (\lambda_1 - a)^2}}, \\ z_{12} = \sqrt{\dfrac{(\lambda_1 - a)^2}{c^2 + (\lambda_1 - a)^2}}. \end{cases}$$

Для отримання координат вектора, що визначає вісь z_2, можна використовувати друге власне значення λ_2, або, що вимагає меншого обсягу обчислень, можна повернути вектор z_1 на кут $\pi/2$; в цьому випадку отримуємо:

$$\begin{cases} z_{21} = -z_{12}, \\ z_{22} = \ \ z_{11}. \end{cases}$$

Якщо при цьому одна з компонент дорівнює нулю, то вихідний базис є оптимальним і ніяких дій робити не потрібно, оскільки вихідні осі і представляють головні компоненти. Тому далі вважаємо, що $z_{11} \neq 0$ і $z_{12} \neq 0$. В цьому випадку, оскільки обидва базису $Y = (\mathbf{y}_1, \mathbf{y}_2)$ і $Z = (\mathbf{z}_1, \mathbf{z}_2)$ ортонормовані, їх компоненти зв’язані співвідношеннями:

$$\begin{cases} \mathbf{z}_1 = \quad \cos\alpha\, \mathbf{y}_1 + \sin\alpha\, \mathbf{y}_2, \\ \mathbf{z}_2 = -\sin\alpha\, \mathbf{y}_1 + \cos\alpha\, \mathbf{y}_2; \end{cases}$$

$$\begin{cases} \mathbf{y}_1 = \cos\alpha\, \mathbf{z}_1 - \sin\alpha\, \mathbf{z}_2, \\ \mathbf{y}_2 = \sin\alpha\, \mathbf{z}_1 + \cos\alpha\, \mathbf{z}_2, \end{cases}$$

де

$$\begin{cases} \cos\alpha = z_{11}, \\ \sin\alpha = z_{12}. \end{cases} \tag{2.14}$$

Відповідно, координати (x'_{1i}, x'_{2i}) і (x_{1i}, x_{2i}) точки $\mathbf{x}_i = (x_{1i}, x_{2i})$ в новому і старому базисах пов’язані такими співвідношеннями

$$\begin{cases} x'_{1i} = \quad \cos\alpha \cdot x_{1i} + \sin\alpha \cdot x_{2i}, \\ x'_{2i} = -\sin\alpha \cdot x_{1i} + \cos\alpha \cdot x_{2i}, \end{cases}$$

а кут повороту, якщо потрібно, можна знайти з (2.14).

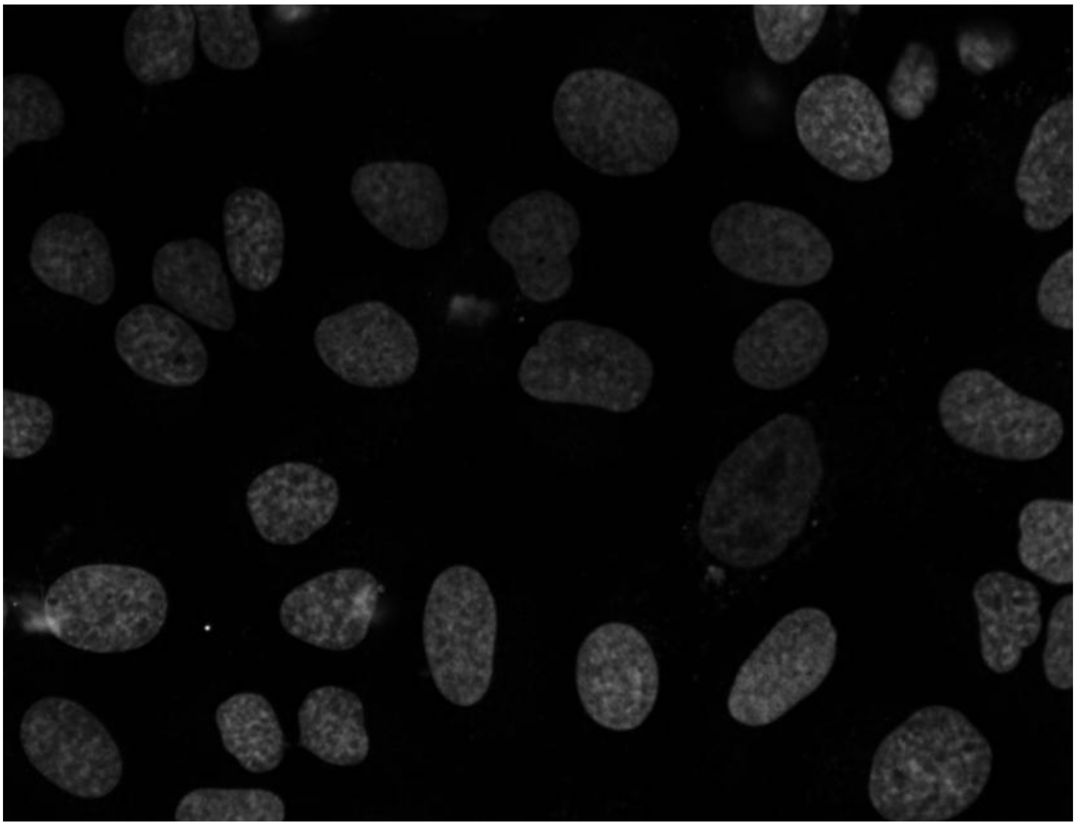
3 РЕАЛІЗАЦІЯ МОДЕЛЕЙ СЕГМЕНТАЦІЇ І ЛОКАЛІЗАЦІЇ ОБ’ЄКТІВ

3.1 Загальна модель сегментації і локалізації об’єктів

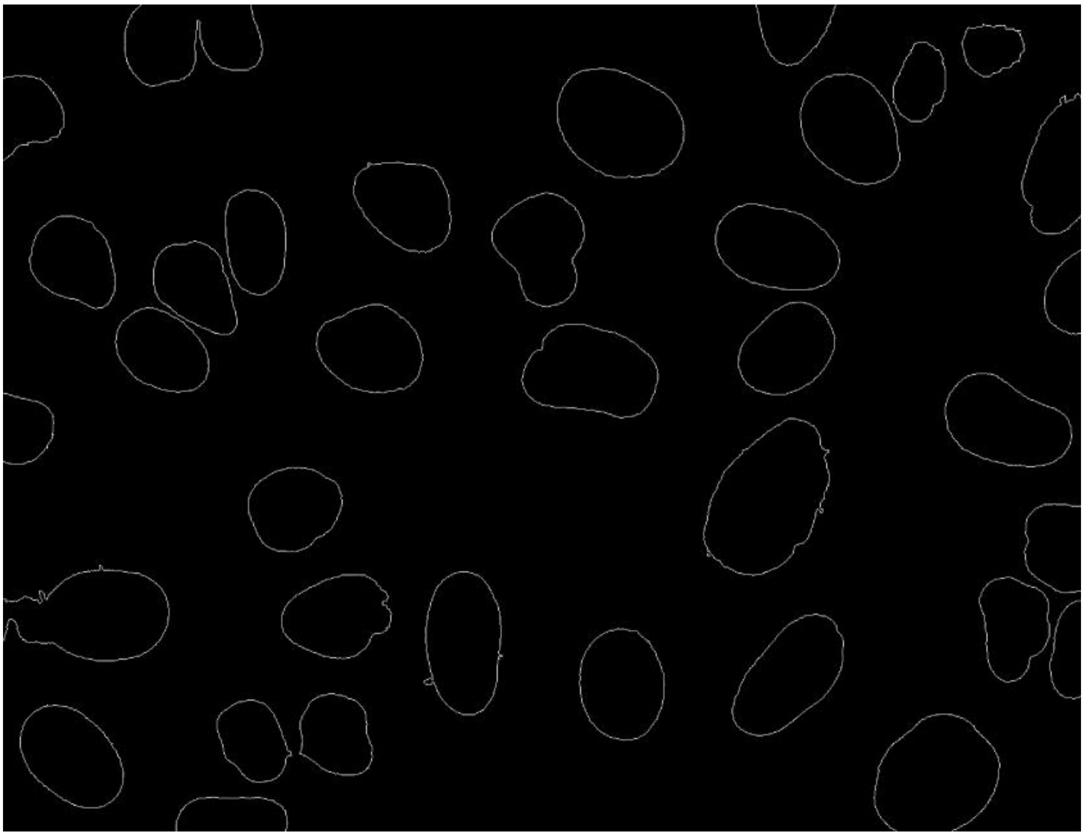
Відповідно до запропонованої моделі перш за все будемо використовувати модель контурної сегментації для знаходження границь об’єктів на цифровій фотографії (рис. 3.1).

Після цього будемо застосовувати до контурів хвильовий метод задля їх зв’язування у окремі зв’язні компоненти границі. При цьому будемо виконувати трасування границі для того, щоб отримувати лише зовнішню границю об’єктів без гілок і петель. Щоб визначити однозв’язну область об’єкту, який розглядається.

Зазначимо, що у багатьох задачах цього достатньо і для локалізації об’єктів на знімку.



а)



б)

Рисунок 3.1 – Вихідна фотографія (а) і контури об’єктів на ній (б)

Якщо це не так, тоді до кожної такої зв’язної компоненти будемо застосовувати метод головних компонент, яких у задачах комп’ютерного зору використовують задля компенсації пороту об’єкту (рис. 3.2), наприклад, в системах розпізнавання обличчя людини.

Застосування цього методу дозволяє адекватно визначити орієнтацію об’єкту у просторі. Після цього мі можемо визначити максимальні і мінімальні координати пік селів об’єкту по осям, знайти відповідні центри і визначити адекватні параметри прямокутника (рис. 3.3).

Після цього координати такого прямокутнику будуть однозначно визначати область локалізації об’єкту на знімку.

Після цього задачу локалізації у тому, чи іншому сенсі можна вважати вирішеною.

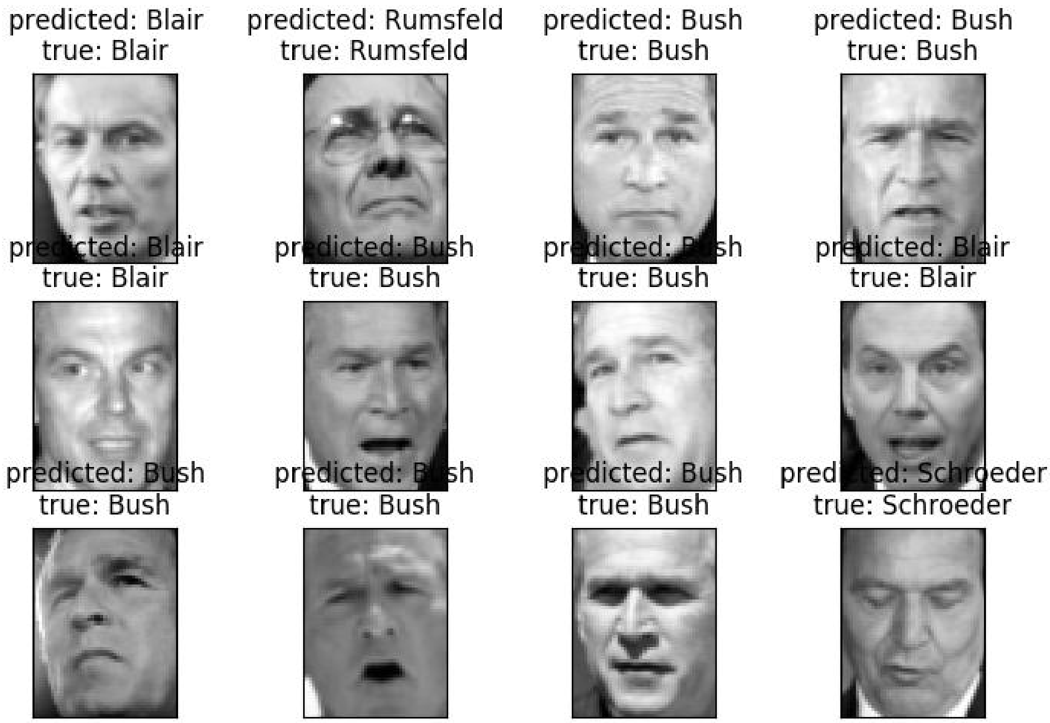


Рисунок 3.2 – Поворот обличчя в системі розпізнавання [21]

```
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.datasets import fetch_lfw_people
from sklearn.metrics import classification_report
from sklearn.decomposition import PCA
from sklearn.neural_network import MLPClassifier


# Load data
lfw_dataset = fetch_lfw_people(min_faces_per_person=100)

_, h, w = lfw_dataset.images.shape
X = lfw_dataset.data
y = lfw_dataset.target
target_names = lfw_dataset.target_names

# split into a training and testing set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

Рисунок 3.3 – Лістинг: Обробка даних е системи Face Recognition [21]

Оскільки відтепер ми знаємо кут повороту зображення, після локалізації при виникненні потреби ми завжди можемо виконати нормалізацію і компенсувати поворот на цей кут (рис. 3.4). Загальна модель локалізації об’єкту на зображенні наведено на рисунку 3.5.

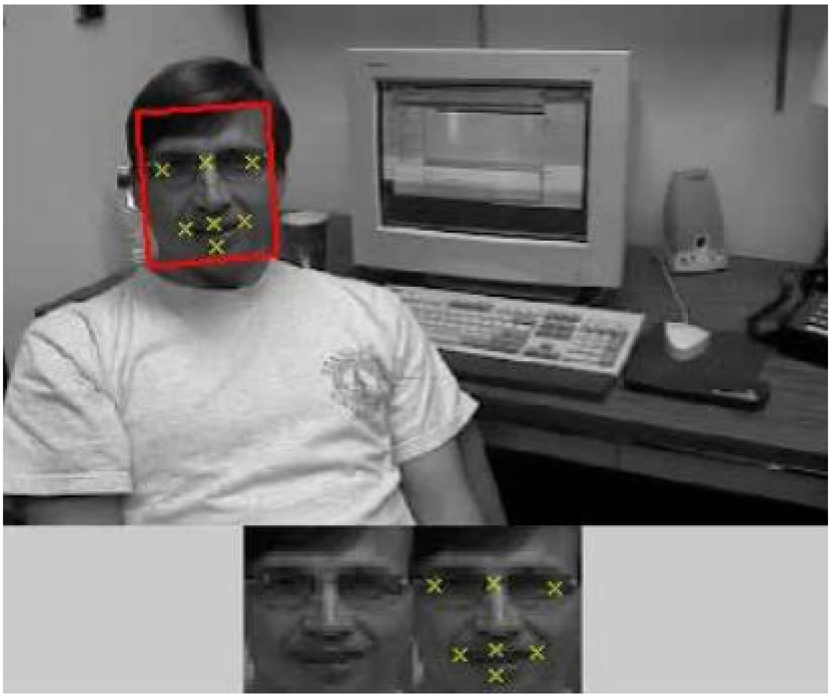


Рисунок 3.4 – Поворот зображення в системі розпізнавання [22] з метою компенсації повороту обличчя

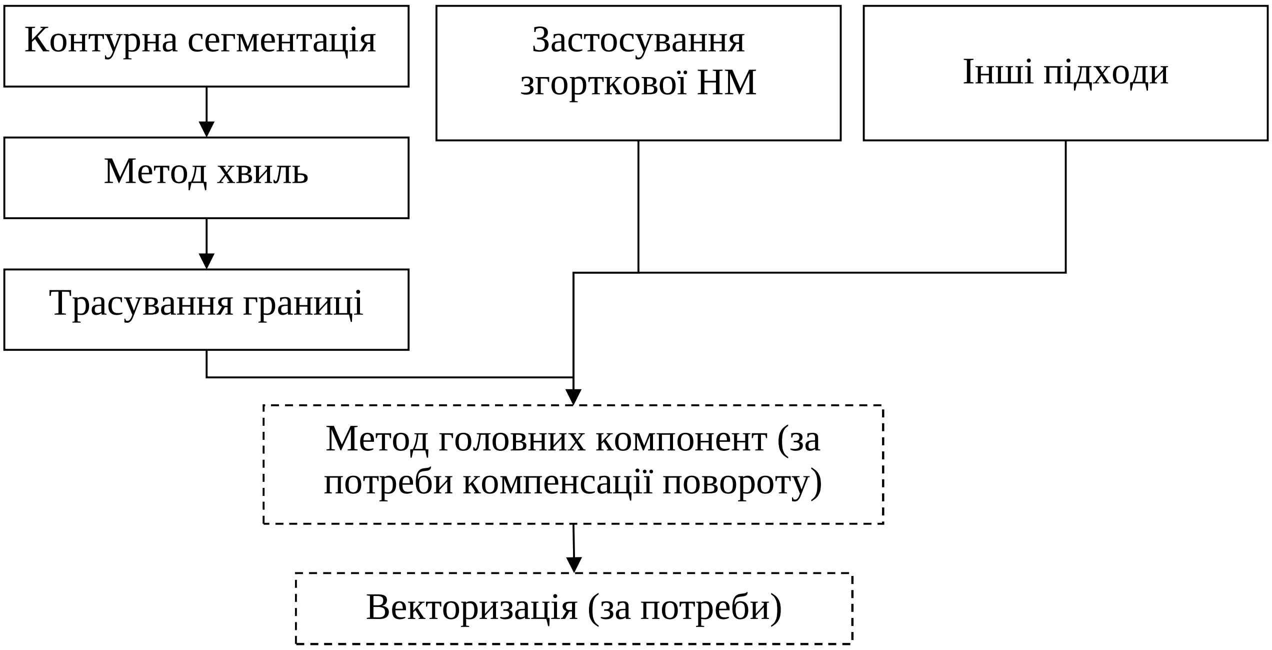


РИСУНОК 3.5 – ЗАГАЛЬНА МОДЕЛЬ ЛОКАЛІЗАЦІЇ ОБ’ЄКТУ

ВИСНОВКИ

В результаті виконання атестаційної роботи отримані наступні основні результати.

Відповідно завданню проведено аналіз сучасного стану питання, досліджені існуючі моделі і підходи до сегментації і локалізації об’єкту на цифровому зображенні з урахуванням трудомісткості, механізмів адаптації до положення границь і варіацій яскравості.

Обрана відповідна система моделей і побудовано алгоритм комплексної сегментації і локалізації об’єкту на цифровому зображенні, з урахуванням функції трасування, яка застосовується задля отримання зовнішньої границі об’єкту без гілок і петель.

Побудовано відповідну загальну модель локалізації, з використанням якої проведений експеримент і показана відповідність реальних і теоретичних оцінок ефективності запропонованого алгоритму.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
2. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – М. : Вильямс, 2004. – 928 с.
3. Журавлев Ю. И. Распознавание. Математические методы. Программная система. Практические применения / Ю. И. Журавлев, В. В. Рязанов, О. В. Сенько. – М. : Фазис, 2005. – 159 с.
4. Шапиро Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман. – М. : БИНОМ, 2006. – 752 с.
5. Смеляков К. С. Модели и методы сегментации изображений объектов нерегулярного вида для автономных систем технического зрения: дис. … доктора техн. наук : 01.05.02 / Смеляков Кирилл Сергеевич. – Х., 2012. – 306 с.
6. Цифровая обработка изображений в информационных системах / [И. С. Грузман, В. С. Киричук, В. П. Косых и др.]. – Новосибирск : НГТУ, 2002. – 351 с.
7. Дуда Р. Распознавание образов и анализ сцен / Р. Дуда, П. Харт. – М. : Мир, 1976. – 230 с.
8. Грицык В. В. Распараллеливание алгоритмов обработки информации в системах реального времени / В. В. Грицык. – К. : Наукова думка, 1981. – 215 с.
9. Sonka M. Image processing, analysis and machine vision / M. Sonka, V. Hlavak, R. Boyle. – California (USA) : Cole Publishing Company, 1999. – 770 p.
10. Яне Б. Цифровая обработка изображений / Б. Яне. – М. : Техносфера, 2007. – 584 с.
11. Смеляков К. С. Модели и методы сегментации границ изображений

нерегулярного вида на основе адаптивных масок: дис. … канд. техн. наук : 01.05.02 / Смеляков Кирилл Сергеевич. – Х., 2005. – 162 с.

12. Шлезингер М. И. Десять лекций по статистическому и структурному распознаванию / М. И. Шлезингер, В. Главач. – К. : Наукова думка, 2004. – 545 с.

13. Машталир В. П. Точечно-множественные методы обработки информации / В. П. Машталир. – Х. : Бизнес Информ, 2001. – 199 с.

14. Методы компьютерной обработки изображений / под ред. В. А. Сойфера. – М. : Физматлит, 2003. – 784 с.

15. Прэтт У. Цифровая обработка изображений. Т. 1 / У. Прэтт. – М. : Мир, 1982. – 312 с.

16. Прэтт У. Цифровая обработка изображений. Т. 2 / У. Прэтт. – М. : Мир, 1982. – 480 с.

17. Путятин Е. П. Обработка изображений в робототехнике / Е. П. Путятин, С. И. Аверин. – М. : Машиностроение, 1990. – 320 с.

18. Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba OBJECT DETECTORS EMERGE IN DEEP SCENE CNNS // ICLR 2015, 15 Apr 2015. – 12p.

19. Overview of the YOLO Object Detection Algorithm [Електронний ресурс] – Режим доступу до ресурсу: https://medium.com/@ODSC/overview-of-the-yolo-object-detection-algorithm-7b52a745d3e0.

20. Метод головних компонент: [Електронний ресурс] – Режим доступу до ресурсу: https://dxdy.ru/topic47566.html.

21. Face Recognition: [Електронний ресурс] – Режим доступу до ресурсу: https://pythonmachinelearning.pro/face-recognition-with-eigenfaces/.

22. Computer Vision [Електронний ресурс] – Режим доступу до ресурсу: https://www.cc.gatech.edu/~afb/classes/CS4495-Fall2014/slides/CS4495-PCA.pdf.