

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Нейромережне розпізнавання об'єктів

Текстова частина до курсової роботи
за спеціальністю „Інженерія програмного забезпечення” 6.050103

Керівник курсової роботи
Ковалюк Т.В.

(прізвище та ініціали)

(підпис)

“ ___ ” _____ 2020 р.

Виконав студент Бондар І.Б.

“ ___ ” _____ 2020р.

Київ 2020

Міністерство освіти і науки України

Зміст

Анотація	2
Вступ	3
ОСНОВНА ЧАСТИНА	5
Розпізнавання об'єктів. Основні принципи.....	5
Метод Віоли-Джонса	8
Розпізнавання номерних знаків.	12
Огляд Інструментів.....	12
K-Nearest Neighbour	12
Згорткові нейронні мережі	13
Open CV	14
Tesseract OCR.....	15
Google Cloud Vision	15
Як це працює	17
Висновки	23
Література.....	25

Анотація

Буде розглянуто можливості та підходи розпізнавання об'єктів за допомогою сучасних технологій та нейромереж. У якості прикладу об'єкту взято номерні знаки автомобілів. Мета – описати та реалізувати алгоритм отримання із статичної фотографії авто текстове представлення його номерного знаку. Ціллю роботи є аналіз існуючих алгоритмів, підходів та методів для реалізації цієї задачі, виведення найкращих та використання їх для реалізації програмного застосунку. Також буде детально описані кроки та методи реалізації задачі.

Вступ

Технологічні можливості сьогодення досить стрімко зростають, порівнюючи з минулими десятиріччями. З ним зростають і можливості ПЗ. Так, розпізнавання об'єктів вже не є чимось дивним та далеким, а можливості програмного забезпечення дозволяють не тільки ідентифікувати об'єкти із статичних зображень, а й робити це у реальному часі, аналізувати ідентифіковані об'єкти та обробляти результати. Наприклад, розпізнавання обличчя людини – вже досить тривіальна задача і навіть розпізнавання емоцій на ньому може зробити звичайний смартфон. Ми всі уже звикли до таких технологій, але як же вони працюють, що коїться «під капотом» та які можливості щодо вдосконалення та додавання нових функціональних можливостей.

Розпізнавання об'єктів - це ключова технологія, яка стоїть за автомобілями без водіїв, що дозволяє розпізнавати знак зупинки або відрізнити пішохода від ліхтаря. Він також корисний у різних сферах застосування, таких як ідентифікація хвороб у біовимірюванні, промислового огляді та роботизованому зорі.

Задача розпізнавання обличчя людини є на сьогодні найпопулярнішою серед задач з розпізнавання об'єктів: в інтернеті є багато матеріалів, прикладів реалізації, початковим кодом під будь-яку платформу та інструмент. Саме тому мене зацікавило вирішення іншої задачі, інформації про яку не так багато, а для коректної та правильної реалізації потрібно детально подумати та проаналізувати існуючі підходи та інструменти.

Розпізнавання номерних знаків автомобілів є наразі дуже актуальною задачею, адже світ рухається до діджиталізацію, поліпшення та автоматизації процесів, які раніше робилися довго та мануально. Програмне забезпечення у комплекті із апаратною частиною (програмно-апаратний комплекс) може не тільки допомогти із автоматизацією процесу паркування авто та сплати за

паркування, але й вирішити задачу із автоматизованим фіксуванням автомобільних правопорушень, їх ідентифікацією та передачею у наступну обробку. Отже, тема є актуальною і доволі цікавою.

Буде розглянуто лише розпізнавання номерних знаків із статичних зображень, адже розпізнавання у реальному часі працює за схожим алгоритмом, але трохи модифікованим, а ціллю цієї роботи є аналіз та огляд інструментів та алгоритмів для реалізації задачі розпізнавання об'єктів.

ОСНОВНА ЧАСТИНА

Розпізнавання об'єктів. Основні принципи

Розпізнавання об'єктів - це техніка комп'ютерного зору для ідентифікації об'єктів у зображеннях чи відео. Розпізнавання об'єктів є ключовим результатом алгоритмів глибокого навчання та машинного навчання. Коли люди дивляться фотографію або дивляться відео, то можуть легко помітити людей, предмети, сцени та візуальні деталі. Мета полягає в тому, щоб навчити комп'ютер робити те, що природно для людини: отримати рівень розуміння того, що містить зображення.

Для розпізнавання об'єктів можна використовувати різноманітні підходи. Останнім часом методи машинного навчання та глибокого навчання стали популярними підходами до розпізнавання предметів. Обидві методики вчать ідентифікувати предмети на зображеннях, але вони відрізняються своїм виконанням.

Технології глибокого навчання стали популярним методом розпізнавання об'єктів. Моделі глибокого навчання, такі як конволюційні нейронні мережі або CNN, використовуються для автоматичного вивчення властивих властивостей об'єкта для ідентифікації цього об'єкта. Наприклад, CNN може навчитися визначати відмінності між котами та собаками, аналізуючи тисячі зображень на тренуваннях та вивчаючи особливості, які роблять котів і собак різними.

Існує два підходи до розпізнавання об'єктів за допомогою глибокого навчання:

Навчання моделі з нуля: для того, щоб навчити глибоку мережу з нуля, треба зібрати дуже великий набір міток даних та спроектувати мережеву архітектуру, яка вивчить функції та побудувати модель. Результати можуть бути вражаючими, але такий підхід вимагає великої кількості даних про

тренування, і потрібно налаштувати шари та ваги в CNN (згортова нейронна мережа).

Використання попередньо вивченої моделі глибокого навчання: Більшість додатків для глибокого навчання використовують підхід передачі навчання - процес, який передбачає точну настройку перевіреної моделі. Ви починаєте з існуючої мережі, наприклад AlexNet або GoogLeNet, і подаєте нові дані, що містять невідомі раніше класи. Цей метод менш трудомісткий і може забезпечити швидший результат, оскільки модель вже пройшла навчання на тисячах або мільйонах зображень.

Щоб виконати розпізнавання об'єктів за допомогою стандартного підходу до машинного навчання, ви починаєте з колекції зображень (або відео) та вибираєте відповідні функції для кожного зображення. Наприклад, алгоритм вилучення можливостей може витягнути крайові або кутові функції, які можна використовувати для розмежування класів у ваших даних. Ці функції додаються до моделі машинного навчання, яка розділить ці особливості на їх окремі категорії, а потім використовувати цю інформацію при аналізі та класифікації нових об'єктів.

Скоріш за все, необхідно буде використовувати різноманітні алгоритми машинного навчання та методи вилучення функцій, які пропонують безліч комбінацій для створення точної моделі розпізнавання об'єктів.

Використання машинного навчання для розпізнавання об'єктів пропонує гнучкість вибору найкращого поєднання функцій та класифікаторів для навчання. Це може досягти точних результатів з мінімальними даними.

Машинне навчання проти глибокого навчання для розпізнавання об'єктів:

Визначення найкращого підходу до розпізнавання об'єктів залежить від програми та проблеми, яку необхідно вирішити. У багатьох випадках машинне навчання може бути ефективною технікою, особливо якщо відомо,

які особливості чи характеристики зображення найкраще використовувати для диференціації класів об'єктів.

Основним моментом, який слід пам'ятати при виборі між машинним та глибоким навчанням - чи є можливість використовувати потужний комп'ютер та чи великий обсяг зображень для тренування. Якщо відповідь на будь-яке з цих питань - «Ні», то найкращим вибором може стати підхід до машинного навчання. Технології глибокого навчання, як правило, краще працюють із більшою кількістю зображень, а GPU (графічний процесор) допомагає скоротити час, необхідний для тренування моделі.

Насправді, функціонал з розпізнання об'єктів складається з трьох підфункціоналів, або етапів, реалізація яких не має бути зв'язана між собою, там може бути виконана різними підсистемами, з використанням різних інструментів і підходів. Головне, що б підсистеми знали формат комунікацію між собою. Це дозволяє обрати найбільш оптимальний інструмент для реалізації кожного з етапів без привязки до одного стеку. Ці три етапи це:

- Detection (виявлення) - знаходження об'єкту на зображенні.
- Recognition (упізнавання) – визнання, отримання інформації про виявлений об'єкт.
- Identification (ідентифікація) – розпізнавання деталей об'єкту, вилучення інформації.



Рисунок 1 Object detection та Object recognition

Метод Віоли-Джонса

Цей алгоритм вважається основним, якщо мова йде про розпізнавання обличчя, окремих його елементів, емоцій. Його було розроблено ще у 2001 році Полом Віолою та Майклом Джонсом, та до сих пір він є основним не тільки для ідентифікації та розпізнавання обличчя, а й для розпізнавання інших об'єктів. Також, його використовують для розпізнавання об'єктів у реальному часі. Дія цього алгоритму побудована на п'яти основних принципах:

- 1) Зображення в інтегральному представленні. Інтегральне представлення зображення – це матриця розмір якої збігається з розміром оригінального зображення. Елементи цієї матриці вираховуються за формулою:

$$I(x,y) = \text{Summ}(I(i,j))$$

Де i та j – яскравість пікселя оригінального зображення. Час на розрахунок інтегрального зображення є лінійно-пропорційним кількості пікселів зображення. Обрахування інтегрального зображення

використовується для швидкого підрахунку заданих частин зображення.

- 2) Використання ознак Хаара. Таку назву вони мають, адже схожі із вейвлетами Хаара. Так, Віола та Джонс, адаптували ідею використання вейвлетів для розпізнавання об'єктів і створили із них ознаки Хаара. Признак Хаара, по своїй суті є декілька суміжних прямокутних частин. Вони певним чином позиціонуються на зображенні, після чого смутуються інтенсивності пікселів у областях для послідуочого вирахування різності між сумами. Ця різність і є «ознакою Хаара»

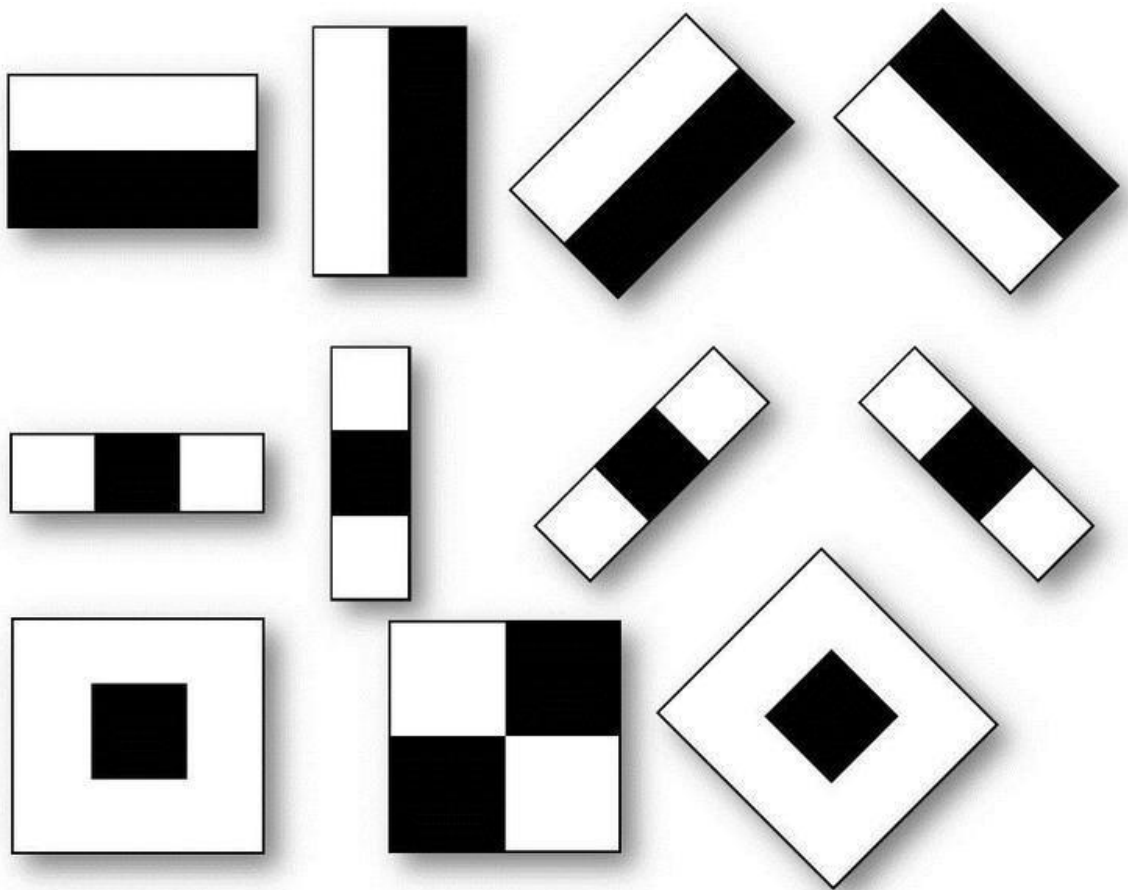


Рисунок 2 Ознаки Хаара

Основною перевагою у використанні ознак Хаара є швидкість цього методу

- 3) Використання бустінгу. Бустінг – це процедура послідовного будування композиції алгоритмів машинного навчання, коли кожен

наступний алгоритм прагне компенсувати недоліки композиції всіх попередніх алгоритмів

- 4) Усі признаки потрапляють у так званий класифікатор, який повертає булеву зміну true або false. Класифікатор – це певний функціонал, що на вхід приймає деяку кількість об’єктів, та «знає» про можливі класи для цих об’єктів. Його задача на виході класифікувати ці об’єкти, тобто розділити їх за певними класами спираючись на певні ознаки.
- 5) Використання каскадів ознак для відкидання частин, де не знайдено необхідний об’єкт. По суті, каскад ознак – це певна група ознак, база для побудови системи виділення складних об’єктів.

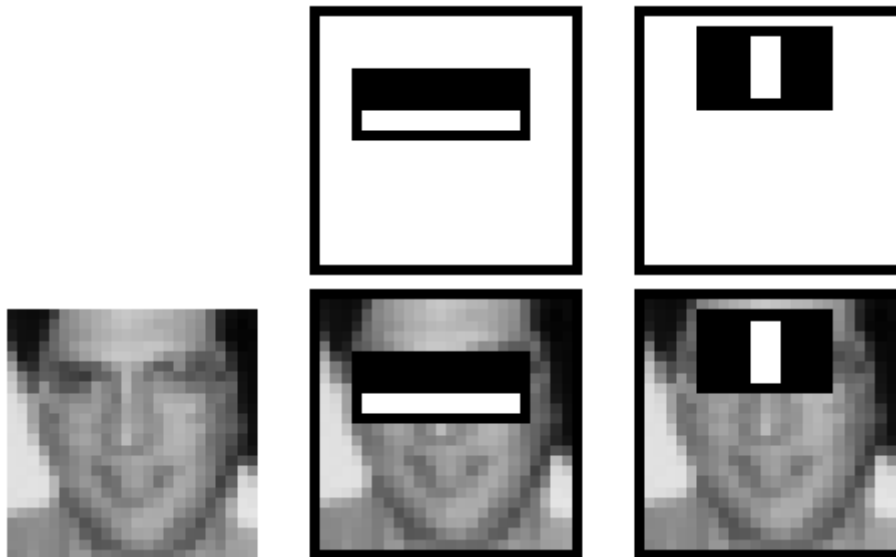


Рисунок 3 приклад використання ознак Хаара у методи Віоли-Джонса

Навчання класифікаторів відбувається дуже повільно, але результати пошуку об’єкту дуже швидкі, саме тому був обраний такий метод розпізнавання об’єктів на зображенні. Це дає змогу отримувати можливість розпізнавання об’єктів навіть у режимі реального часу. Метод Віоли-Джона - один з найкращих за співвідношенням ефективності розпізнавання / швидкості. Також цей детектор має надзвичайно низьку ймовірність помилки при виявленні та ідентифікації обличчя. Алгоритм добре працює і розпізнає деталі об’єкту (такі, як риси обличчя) навіть під невеликим кутом,

приблизно до 30 градусів. При куті нахилу більше 30 градусів відсоток виявлення різко падає. І це не дозволяє стандартній реалізації виявити повернене обличчя людини під довільним кутом, що значно ускладнює або унеможлиблює використання алгоритму в сучасних виробничих системах з урахуванням їх зростаючих потреб.

Необхідний детальний аналіз принципів, на яких базується алгоритм Віола-Джонса. Функціонал цього методу шукає об'єкти та їх деталі за методом «скануючого вікна».

Алгоритм скануючого вікна з ознаками виглядає так:

- є досліджуване зображення, вибрано вікно сканування, обрані використовувані ознаки;
- далі вікно сканування починає послідовно рухатися по зображенню з кроком в 1 осередок вікна (припустимо, розмір самого вікна є 32 x 32 осередки);
- при скануванні зображення в кожному вікні обчислюється приблизно 200 000 варіантів розташування ознак, за рахунок зміни масштабу ознак і їх положення в вікні сканування;
- сканування проводиться послідовно для різних масштабів;
- масштабується не саме зображення, а скануючий вікно (змінюється розмір осередку);
- всі знайдені ознаки потрапляють до класифікатору, який «виносить вердикт».

В процесі пошуку обчислювати всі ознаки на малопотужних настільних ПК просто нереально. Отже, класифікатор повинен реагувати тільки на певний, потрібну підмножину всіх ознак. Абсолютно логічно, що треба навчити класифікатор знаходженню осіб з даного певного підмножині. Це можна зробити, навчаючи обчислювальну машину автоматично.

Розпізнавання номерних знаків.

Алгоритм Віоли-Джонса – це потужний інструмент для розпізнавання об'єктів. Але чи підходить він для всіх задач? Чи є він повністю універсальним? Відповісти на це питання однозначно – не можливо. Так, для розпізнавання номерних знаків явно цей алгоритм не використовується. Але при реалізації функціоналу на різних етапах можуть бути використанні певні кроки із цього алгоритму.

Знаходження об'єктів на зображенні або в відео-поточці це завдання з області комп'ютерного зору, яка вирішується різними підходами, але найчастіше за допомогою, так-званих, згортальних нейронних мереж. Нам потрібно знайти не просто область на фото в якій зустрічається шуканий об'єкт, але і відокремити все його точки від інших об'єктів або фону.

Огляд Інструментів

K-Nearest Neighbour

K Найближчий сусід - це простий алгоритм, який зберігає всі наявні випадки та класифікує нові дані або реєстр на основі міри подібності. Він здебільшого використовується для класифікації точки даних на основі класифікації її сусідів.

kNN Algorithm

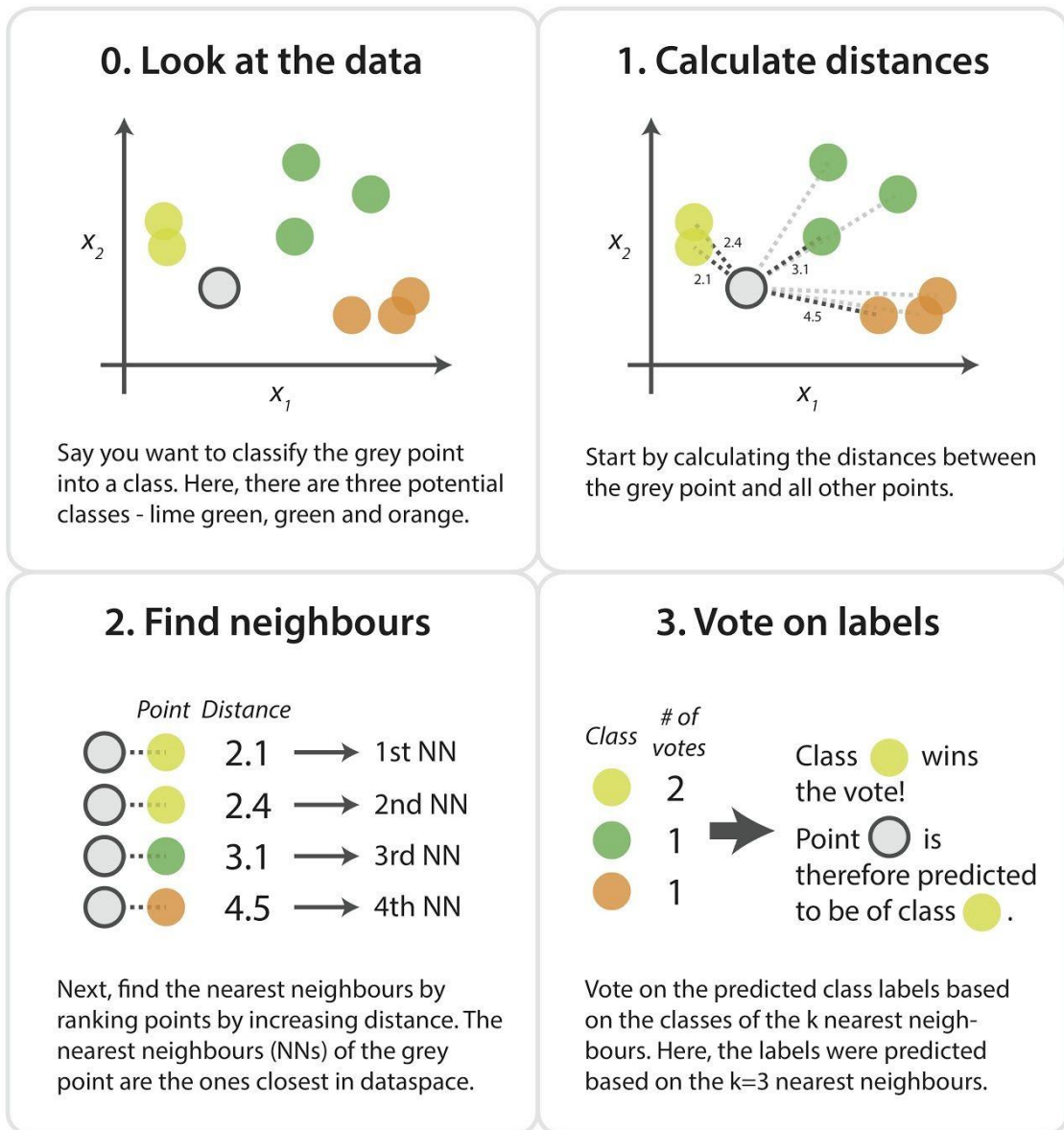


Рисунок 4 Демонстрація принципу роботи алгоритму пошуку найближчого сусіда

Згорткові нейронні мережі

Згорткові нейронні мережі (ЗНМ, CNN) це- одні з найвпливовіших інновацій в області комп'ютерного зору. Вперше нейронні мережі привернули загальну увагу в 2012 році, коли Алекс Крижевський завдяки їм виграв конкурс ImageNet (грубо кажучи, це щорічна олімпіада по машинному зору), знизивши рекорд помилок класифікації з 26% до 15%, що тоді стало

проривом. Сьогодні глибинне навчання лежить в основі послуг багатьох компаній: Facebook використовує нейронні мережі для алгоритмів автоматичного створення тегів, Google - для пошуку серед фотографій користувача, Amazon - для генерації рекомендацій товарів, Pinterest - для персоналізації домашньої сторінки користувача, а Instagram - для пошукової інфраструктури.

Коли комп'ютер бачить зображення (приймає дані на вхід), він бачить масив пікселів. Залежно від дозволу і розміру зображення, наприклад, розмір масиву може бути $48 \times 48 \times 3$ (де 3 - це значення каналів RGB). Щоб було зрозуміліше, давайте уявимо, у нас є кольорове зображення у форматі JPG, і його розмір 500×500 . Відповідний масив буде $500 \times 500 \times 3$. Кожному з цих чисел присвоюється значення від 0 до 255, яке описує інтенсивність пікселя в цій точці. Ці цифри, залишаючись безглуздими для нас, коли ми визначаємо що на зображенні, є єдиними вступними даними, доступними комп'ютера. Ідея в тому, що ви даєте комп'ютера цю матрицю, а він виводить числа, які описують ймовірність класу.

Open CV

Бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення Open CV. Насьогодні є основним інструментом і використовується у майже всіх реалізаціях ПЗ з розпізнавання об'єктів. У неї входять понад 2500 алгоритмів, в яких є як класичні, так і сучасні алгоритми для комп'ютерного зору і машинного навчання. Написана на C++, але має обгортки, що дозволяють використовувати цю бібліотеку на різних платформах і мова. Open CV має величезний функціонал, що задовільняє більшості потреб при роботі із зображенням. Саме тут вже є реалізація CNN, яку необхідно просто адаптувати під необхідні вимоги та використовувати.

Open CV містить не тільки прості функції для завантаження та редагування зображення, а також і більш складні, з елементами машинного навчання,

використання нейронних мереж, тощо. Ця бібліотека є досить базовою для всіх реалізацій.

Tesseract OCR

Оптичне розпізнавання символів (англ. Optical Character Recognition - OCR) - це технологія, яка дозволяє перетворювати різні типи документів, такі як відскановані документи, PDF-файли або фото з цифрової камери, в редаговані формати з можливістю пошуку.

Для розпізнавання символів на номерному знаку автомобіля (після його знаходження на зображенні та ідентифікації) необхідно використовувати саме цю технологію. Лідером, серед функціональних реалізацій цієї технології є ПЗ Tesseract від компанії Hewlett-Packard, розробленої ще в далекі 80-90-ті роки минулого сторіччя. У 2006 це програмне забезпечення було викуплено компанією Google. Наразі, це безпрецедентний лідер серед OCR продуктів, що має не лише високу якість а точність при роботі із текстом, а й можливість роботи з UTF-8. Тобто повна підтримка англійської, російської та навіть української мови за допомогою додаткових модулів.

Google Cloud Vision

Cloud Vision API дозволяє розробникам легко інтегрувати функції виявлення зору в додатки, включаючи маркування зображень, виявлення обличчя та орієнтирів, оптичне розпізнавання символів (OCR) та тегування явного вмісту. Це дуже потужний інструмент, можливості якого важко переоцінити. Він містить у собі багато з вище перерахованих інструментів, працює за методом API що робить можливість взаємодіяти із ним з будь-якої платформи та використовуючи будь яку мову програмування. Великим плюсом є швидкість – зображення обробляється на хмарних серверах Гугл. Зазвичай, запит за часом не перевищує 300 мс.

Приклад використання Google Cloud Vision на Python для ідентифікації та розпізнавання автомобільного номеру.

Для початку необхідно встановити пакет google-cloud-vision та імпортувати його в файл:

```
from google.cloud import vision_v1p3beta1 as vision
```

Далі необхідно створити та вивантажити API ключ з Google Cloud Platform. Для цього необхідно створити новий проект, перейти у розділ IAM та обрати опцію «сервісні акаунти». Після цього необхідно створити нового користувача з правами на ML та сформувавши ключ для цього користувача. Ключ буде сформовано та автоматично завантажено у форматі JSON. Його необхідно додати до проекту та проініціалізувати назвою JSON файлу зміну:

```
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = 'client_key.json'
```

Далі можна створювати новий екземпляр класу клієнту:

```
googleCloudVisonClient = vision.ImageAnnotatorClient()
```

Це майже все! Все що залишилося – завантажити зображення, переформатувати його у об'єкт, який зможе прийняти клієнт та відправити запит:

```
with io.open(img_path, 'rb') as image_file:
    content = image_file.read()

    image = vision.types.Image(content=content)

    response = googleCloudVisonClient.text_detection(image=image)
```

В результаті виконання запиту хмарна платформа поверне нам усі знаходження тексту на зображенні. Залишається лише обрати потрібні. У випадку із розпізнанням номеру авто – достатньо налаштувати фільтрацію вихідних стрічок за певним параметром (наприклад, якщо мова йде тільки про українські номери – то номер має містити 8 символів та 4 з них в середині мають бути цифрами. Плюс номера-винятки).

Як це працює

Використовуючи Google Cloud Vision – можна досягти нереальних результатів. Але як це насправді працює? Що відбувається «під капотом» у цього API? Як виглядає реалізація функціоналу без використання сторонніх сервісів? Що ж, давайте розбиратися.

Приклад реалізації функціоналу розпізнавання автомобільних номерів з використанням мови програмування Python та бібліотек numpy і Open CV.

Спочатку необхідно отримати вхідне зображення.

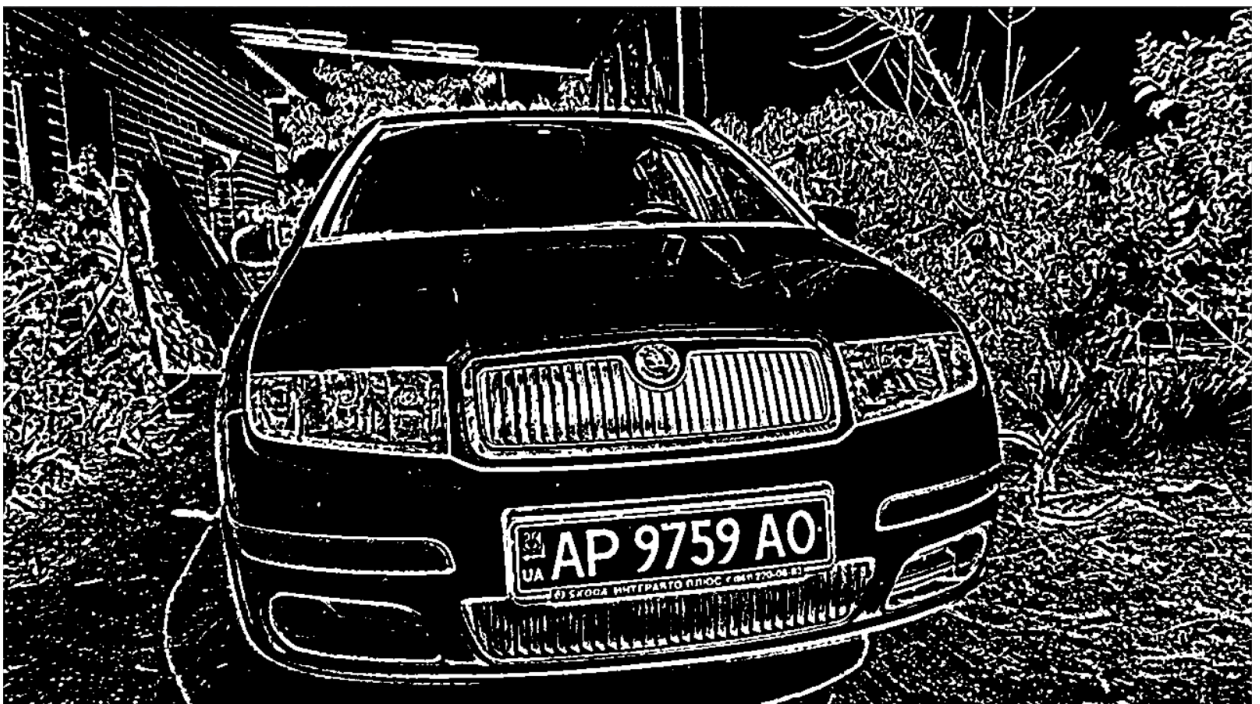


Рисунок 5 Оригінальне зображення без обробки

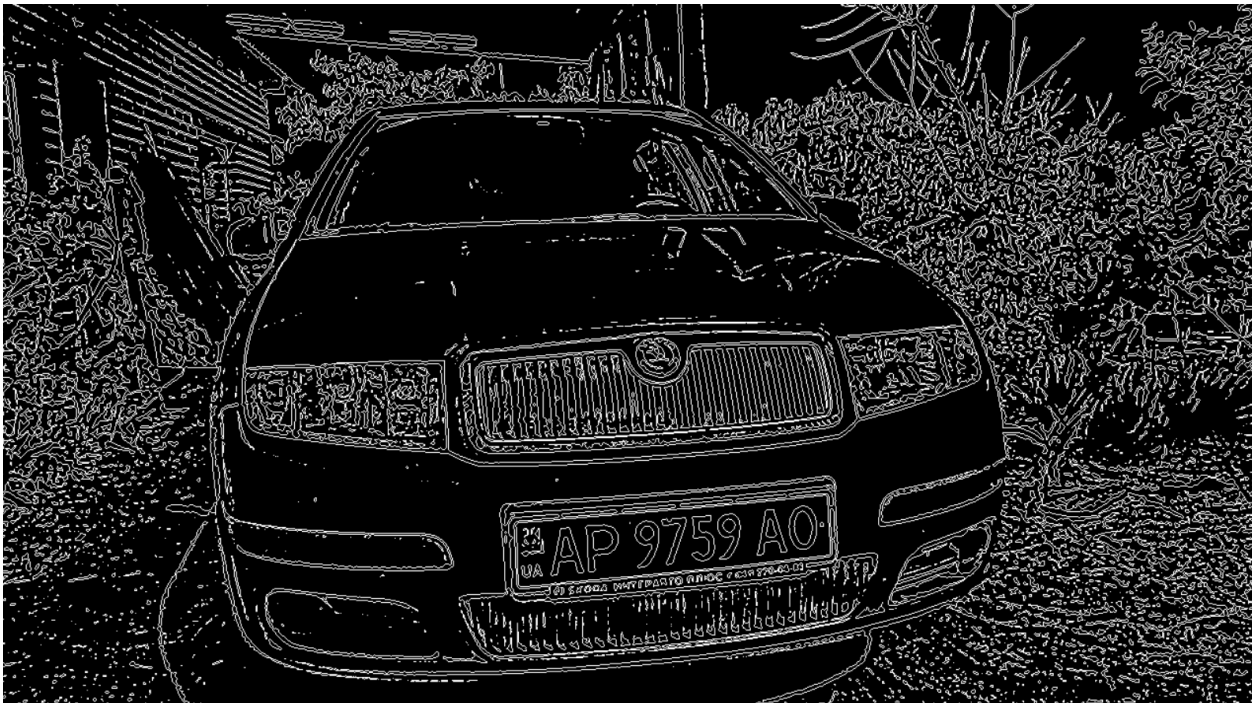
Далі, необхідно виділити більш світлі та більш темні місця. Тобто отримати зображення у сірому представленні. По суті, ця обробка, та деякі інші за нею просто дозволяють виділити на зображеннях цікаві області, без їх аналізу. Велика частина цих методів застосовує якесь єдине перетворення на всі точки зображення. На рівні фільтрації аналіз зображення не проводиться, але точки, які проходять фільтрацію, можна розглядати як області з особливими характеристиками.



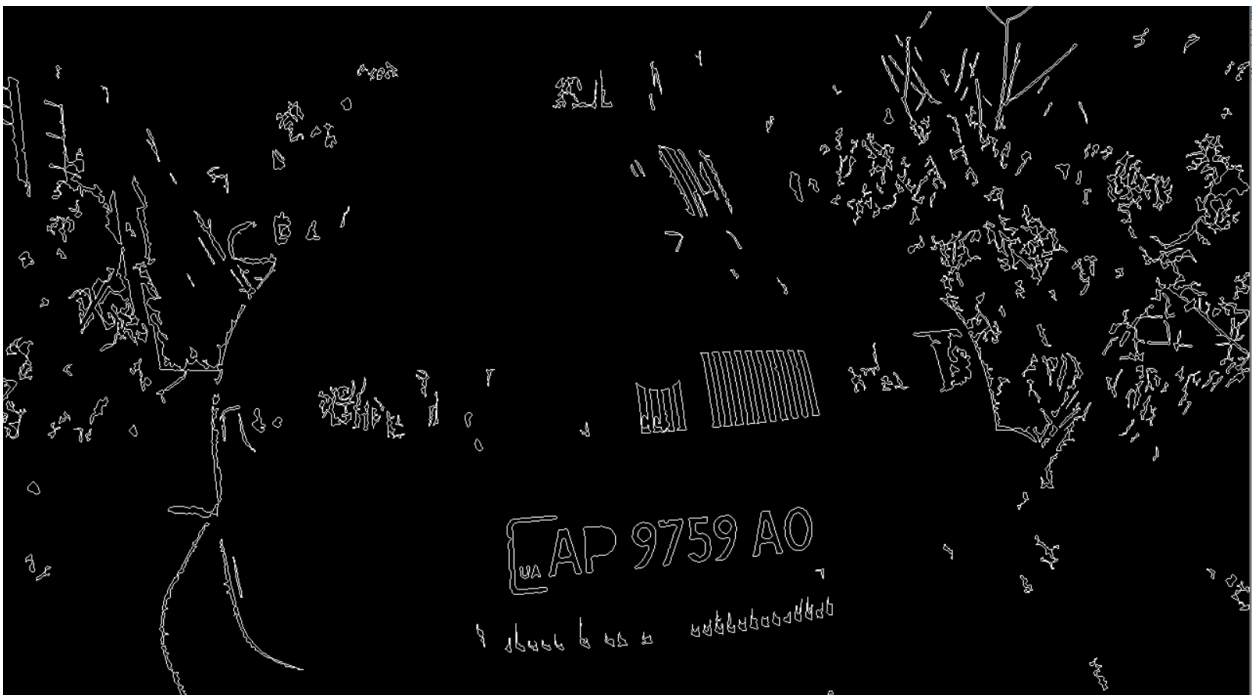
Після цього, використовуючи попереднє зображення, можна отримати зображення так званого trashScene.



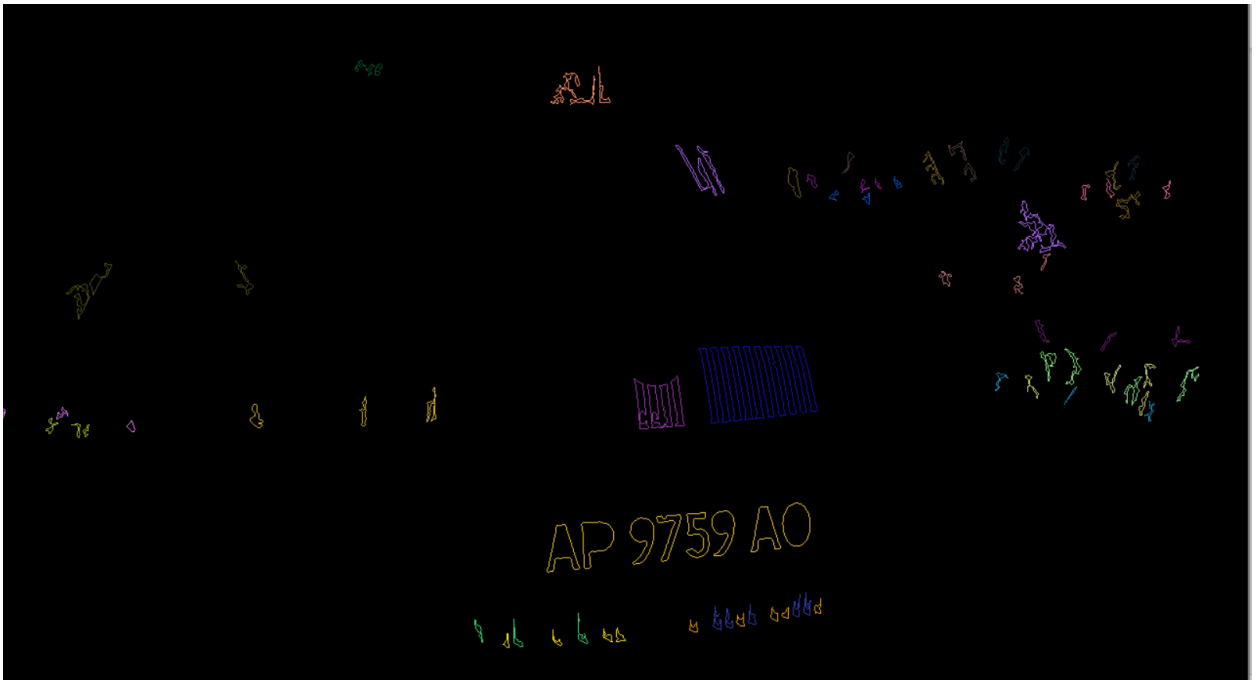
Далі, за допомогою методу з бібліотеки Open CV необхідно отримати зображення усіх можливих контурів та перевірити, чи можливо, що вони є символами, чи ні. Ті, що не пройшли перевірку на чи є символом не мають потрапити у наступне зображення



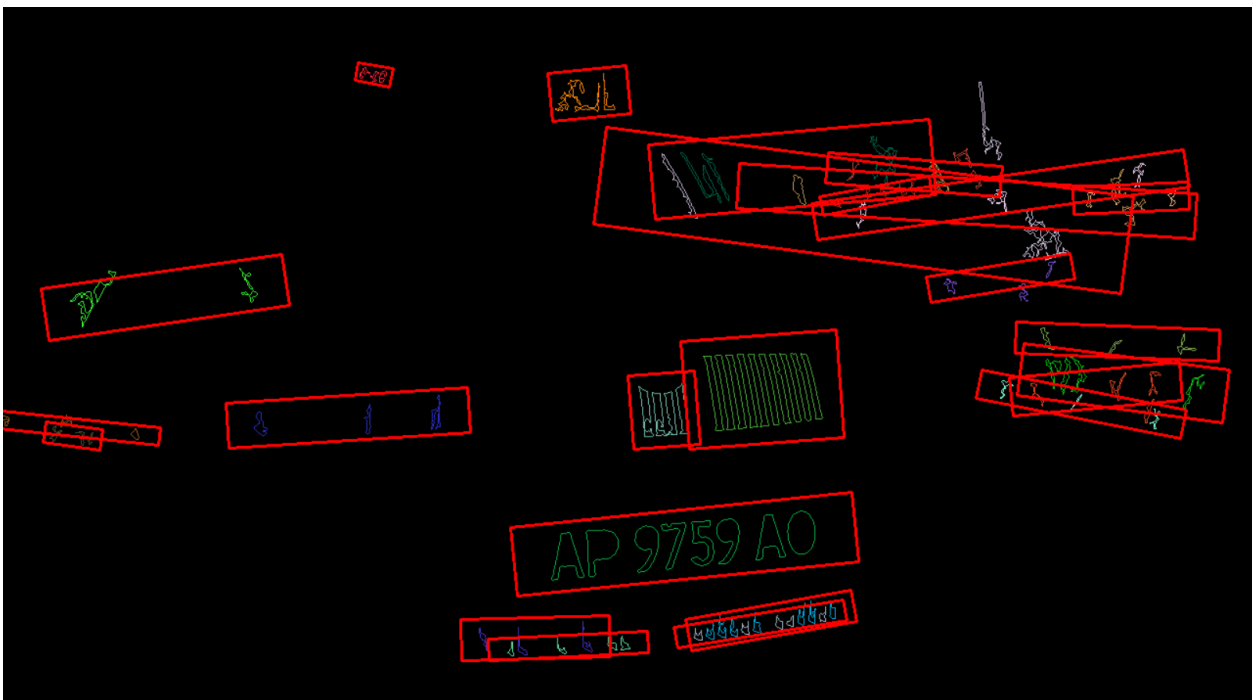
Далі, за допомогою методу отримання векторів, необхідно знайти контурне представлення попереднього зображення



Після цього, необхідно згрупувати елементи, що залишилися у групи можливих символів. Параметр групування це алгоритм найближчого сусіда. Я видно на зображенні нижче, для наглядності можливі символи (точніше, їх групи) розфарбовано різними кольорами. Серед них і справжній номер.



На цьому процес Detection завершено – було знайдено всі об’єкти (а точніше їх групи) що потенційно можуть бути символами. Далі, час стадії Recognition – для кожні групи буде використано алгоритм для виявлення з неї символів.



Тут все доволі схоже з початковою обробкою оригінального зображення: за координатами береться група потенційних символів ,а саме їх частина з оригінально зображення та виконуються схожі кроки для відкидання всього зайвого:



На цьому кроці кожен символ обробляється окремо. Береться його координати, символ «вирізається» з зображення. Потім це зображення сплющується у одновимірний масив за допомогою бібліотеки `numpy` та викликається функція з машинного навчання по пошуку найближчого сусіда. Звісно, мережу треба натренувати, перед її використанням. Чим більше

тестових даних буде використано для тренування, тим чіткіші будуть прогнози та розпізнання символів.



```
license plate read from image = AP9759AO
```

Так, на виході маємо текст з номерного знаку у текстовому представленні. Тобто, всі частини алгоритму відпрацювали успішно.

Як видно з приклада вище – розпізнавання такого об’єкту як номерний знак і отримання інформації з нього у текстовому представленні – це доволі об’ємний процес з використанням багатьох різних підходів і методик на різних його етапах.

Висновки

Сьогодні, задача з розпізнавання об'єктів є дуже актуальною, і не дивлячись на те, що більшість алгоритмів і підходів до її вирішення було розроблено багато років тому, вона продовжує стрімко набирати актуальність у різних сферах нашого життя. У комбінації з машинним навчанням та нейронними мережами, можливості програмного забезпечення з розпізнавання об'єктів є досить широкими і продовжують зростати.

Було оглянуто не мало алгоритмів і інструментів, що домагають у створенні якісного програмного забезпечення під будь-яку задачу. До того ж, було виявлено, що таких інструментів навіть для одного функціоналу чимало, і треба проводити детальний аналіз для виявлення найбільш підходящого під конкретну задачу.

Основним інструментом для створення програми з розпізнавання об'єктів є бібліотека Open CV – не зважаючи на те, що вона була створена доволі давно, розробники не зупиняють її оновлення, вона має величезні можливості для роботи із зображеннями та їх редагуванням.

Також було виявлено, що використання нейромереж, тренування моделей і їх використання у задачі з розпізнавання об'єктів сильно поліпшує роботу розробнику і допомагає досягнути поставлених цілей набагато простіше.

Через велику популярність задачі з розпізнавання об'єктів, сьогодні вже є багато реалізованих функціоналів, що безперечно є лідерами на ринку такого ПЗ. Наприклад, хмарний сервіс машинного навчання від Google. Мною було відтворено та відтестовано чимало різних підходів та алогоритмів, та використання цього сервісу сподобалося найбільше – він найточніше розпізнавав текст на номерному знаку автомобіля, отримав перемогу у швидкості та, що не менш важливо – у складності реалізації. Можна сказати, що цей сервіс працює за метдом black box – тобто ми знаємо що має бути на вхід і що на вихід – а що відбувається у ньому то не наша справа. Цікаво, що

саме так працюють і нейронні мережі, тому, можливо використання сторонніх сервісів не так і погано? До того ж, навіть винаходити велосипед, коли вже є потужна і сильна його реалізація?

У будь якого випадку, наразі тема є дуже актуальною: розпізнавання об'єктів використовують не тільки для розпізнавання обличчя або номерного знаку автомобіля, алей й у медицині (адже у комбінації із машинним навчанням, з'явилася можливість розпізнавати хвороби за певними симптомами, тощо), будівництві, альтернативних джерелах енергії, тощо.

Література

- 1) Стаття “Beginner’s Guide to Object Detection Algorithms”:
<https://medium.com/analytics-vidhya/beginners-guide-to-object-detection-algorithms-6620fb31c375>
- 2) Ресурс з машинного навчання:
<http://www.machinelearning.ru/>
- 3) Офіційна сторінка Open CV:
<https://opencv.org/about/>
- 4) Зображення роботи алгоритму пошуку найближчого сусіда:
<https://www.pinterest.com/pin/817755244810706876/>
- 5) Стаття про розпізнавання автомобільних номерів:
<https://habr.com/ru/post/439330/>
- 6) Стаття “Automatic License Plate Detection & Recognition using deep learning”
<https://towardsdatascience.com/automatic-license-plate-detection-recognition-using-deep-learning-624def07eaaf>
- 7) Книга “Изучаем OpenCV 3” (Адріан Келер, Гєри Бредскі, 2017 р.)
- 8) Стаття про Tesseract з ресурсу Вікіпедія:
<https://ru.wikipedia.org/wiki/Tesseract>