

Технічні науки

УДК 004.8

**Іванюк Володимир Ігорович**

*студент*

*Національного технічного університету України  
«Київський політехнічний інститут імені Ігоря Сікорського»*

**Ivaniuk Volodymyr**

*Student of the*

*National Technical University of Ukraine  
"Igor Sikorsky Kyiv Polytechnic Institute"*

**Потапова Катерина Романівна**

*кандидат технічних наук, доцент*

*Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»*

**Potapova Kateryna**

*Candidate in Technical Sciences, Associate Professor*

*National Technical University of Ukraine  
"Igor Sikorsky Kyiv Polytechnic Institute"*

**Наливайчук Микола Васильович**

*кандидат технічних наук, старший викладач кафедри СПіСКС*

*Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»*

**Nalyvaychuk Mykola**

*Candidate in Technical Sciences, Senior Lecturer*

*National Technical University of Ukraine  
"Igor Sikorsky Kyiv Polytechnic Institute"*

**ОГЛЯД НЕЙРОННИХ МЕРЕЖ ТА СТАНДАРТНИХ АЛГОРИТМІВ  
ОБРОБКИ ЗОБРАЖЕНЬ В ОБЛАСТІ КОМП'ЮТЕРНОГО ЗОРУ  
SURVEY OF NEURAL NETWORKS AND STANDARD IMAGE  
PROCESSING ALGORITHMS IN THE FIELD OF COMPUTER VISION**

***Анотація.** Дане дослідження присвячене важливому питанню порівняння стандартних алгоритмів комп'ютерного зору із нейронними мережами. Стандартні алгоритми, основані на традиційних методах обробки зображень та комп'ютерного бачення, вже довгий час знаходять широке застосування в різних галузях. З іншого боку, нейронні мережі, інспіровані структурою людського мозку, зазнали бурхливого розвитку і стали головним інструментом в глибокому навчанні, аналізі великих обсягів даних та обробці зображень.*

*Питання, яке ця стаття ставить перед собою, - це як обрати оптимальний підхід до розв'язання завдань комп'ютерного зору. Стандартні методи, з їхніми класичними та перевіреними рішеннями, чи нейронні мережі, які демонструють вражаючу здатність до автоматичного вивчення складних залежностей у даних. Мета цієї статті – розглянути це питання з різних сторін, аналізуючи переваги та обмеження обох підходів.*

***Ключові слова:** комп'ютерний зір, нейронні мережі, алгоритми обробки зображень.*

***Abstract.** This research focuses on the important issue of comparing standard computer vision algorithms with neural networks. Standard algorithms based on traditional methods of image processing and computer vision have long been widely used in various industries. On the other hand, neural networks, inspired by the structure of the human brain, have undergone rapid development and have become a major tool in deep learning, big data analysis, and image processing.*

*The question this article poses is how to choose the optimal approach to solving computer vision problems. Standard methods, with their classical and proven solutions, or neural networks, which demonstrate an impressive ability to automatically learn complex dependencies in data. The purpose of this article is to consider this issue from different angles, analysing the advantages and limitations of both approaches.*

**Key words:** *computer vision, neural networks, image processing algorithms.*

**Вступ.** В останні десятиліття спостерігається вразливий розвиток області комп'ютерного зору, яка виявляє потенціал революційної трансформації в багатьох аспектах нашого повсякденного життя. Із зростанням доступності обчислювальних ресурсів та збільшенням об'єму даних, що обробляються, постає необхідність вдосконалення методів обробки та аналізу зображень. В цьому контексті питання вибору оптимальних алгоритмів для досягнення максимальної точності та продуктивності набуває особливої актуальності.

Стандартні алгоритми комп'ютерного зору, які базуються на класичних математичних та обчислювальних принципах, вже довгий час служать основою для багатьох застосувань, починаючи від відстеження об'єктів на відеозаписах до автоматизованої медичної діагностики. З іншого боку, нейронні мережі, інспіровані структурою людського мозку, набули великого успіху в багатьох галузях завдяки їхній здатності до автоматичного вивчення складних залежностей у даних.

В даній роботі ми розглянемо основні принципи і особливості обох підходів, порівняє їх переваги та недоліки та підкреслить області, де кожен з них виявляється найбільш ефективним. Наша мета - зробити широкий огляд сучасних досягнень в галузі обробки зображень та комп'ютерного

зору, щоб зрозуміти, які підходи визначають подальший розвиток цієї науки та як вони впливають на наше повсякденне життя та індустріальний прогрес.

### Матеріали та методи.

Детектор Канні є одним із найбільш широко використовуваних алгоритмів виявлення меж у зображеннях. Він був розроблений Джоном Канні в 1986 році [1] і досі залишається популярним завдяки своїй ефективності та точності.

Процес алгоритму виявлення контурів Канні можливо розбити на п'ять різних етапів:

1. Згладжування (Smoothing): Застосовується фільтр Гауса для згладжування зображення і зниження шуму. Оператор Канні використовує фільтр (1), який може бути добре наближений до першої похідної гауссіани  $\sigma = 1.4$ .

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A \quad (1)$$

де  $A$  – це вхідне зображення,  $B$  – вихідне зображення.

2. Обчислення градієнта (Gradient Calculation): Обчислюються градієнти яскравості в зображенні з використанням оператора Собеля. Межі відзначаються там, де градієнт зображення набуває максимального значення. Вони можуть мати різний напрямок, тому алгоритм Канні використовує чотири фільтри для виявлення горизонтальних, вертикальних і діагональних ребер у розмитому зображенні. Оператор використовує ядра  $3 \times 3$ , які згорнуті з вихідним зображенням для обчислення наближень похідних - одне для горизонтальних змін, а інше для вертикальних. Нехай  $A$  - це вихідне зображення, а  $G_x$  і  $G_y$  - два зображення, на яких кожна точка містить наближені похідні по  $x$  і по  $y$ . Вони обчислюються таким чином:

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \text{ та } G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \quad (2)$$

Кут напрямку вектора градієнта округлюється і може набувати таких значень: 0, 45, 90, 135.

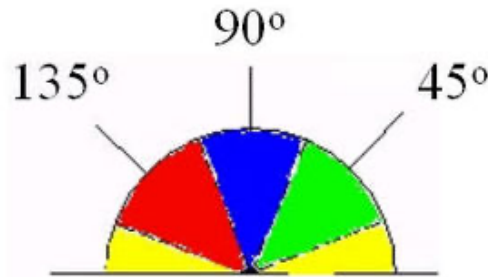


Рис. 1. Схематичне зображення округлень значень ребер

3. Придушення не-максимумів (Non-maximum Suppression): Кордони уточнюються шляхом придушення всіх пікселів, які не є локальними максимумами в напрямку градієнта.

4. Подвійний поріг (Double Thresholding): Застосовується дворівневе порогове значення для визначення "сильних" і "слабких" кордонів.

5. Остаточне визначення межі (Edge Tracking by Hysteresis): Сильні межі зберігаються, а слабкі межі зберігаються тільки в тому разі, якщо вони з'єднуються із сильною межею.

Детектор Канні з високою точністю виявляє краї об'єктів на зображенні, але при цьому вимогливий до обчислювальних ресурсів і не завжди ефективний у випадках з високими шумами або складним фоном.

Загалом, детектор Канні є корисним інструментом для задач виявлення об'єктів та аналізу контурів на зображеннях, і його ефективність доповнюється іншими алгоритмами обробки зображень для отримання точних та надійних результатів.

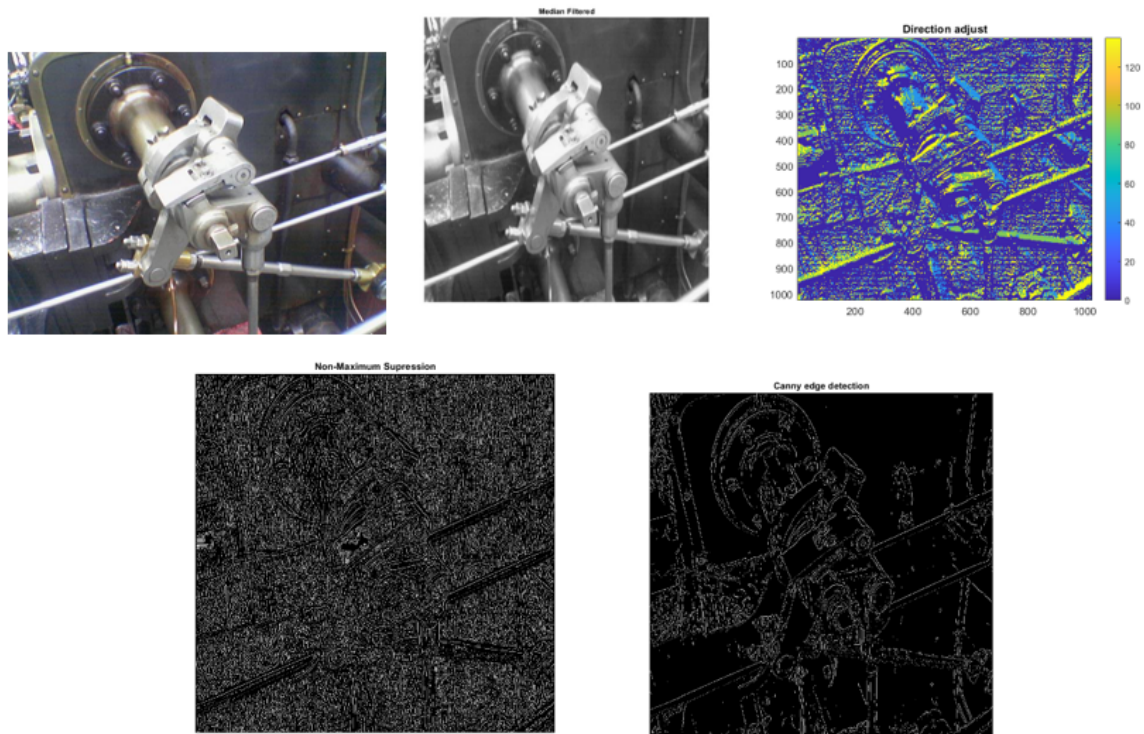


Рис. 2. Поетапне створення границь за допомогою детектора Канні

**Перетворення Хафа** (Hough transform, HT) був представлений як спосіб виявлення складних патернів точок у бінарних зображеннях. Його сутність полягає у визначенні конкретних параметрів, які описують ці патерни. Потім просторово розтягнуті патерни піддаються трансформації так, щоб вони утворювали компактні об'єкти у просторі можливих значень параметрів. За допомогою цього методу складну глобальну задачу виявлення у зображенні перетворюють на більш просту задачу виявлення локальних піків у просторі параметрів.

Основні ідеї цього методу можна проілюструвати на прикладі визначення множини точок на зображенні, які лежать на одній прямій (колінеарні точки). Цю множину точок зображення  $(x, y)$ , що лежать на одній прямій, можна описати відношенням  $f$ , таким, що

$$f = ((\hat{m}, \hat{c}), (x, y)) = y - \hat{m}x - \hat{c} = 0, \quad (3)$$

де  $m$  і  $c$  - два параметри, нахил і перетин, які характеризують лінію. Рівняння (1) ставить у відповідність кожному значенню комбінації

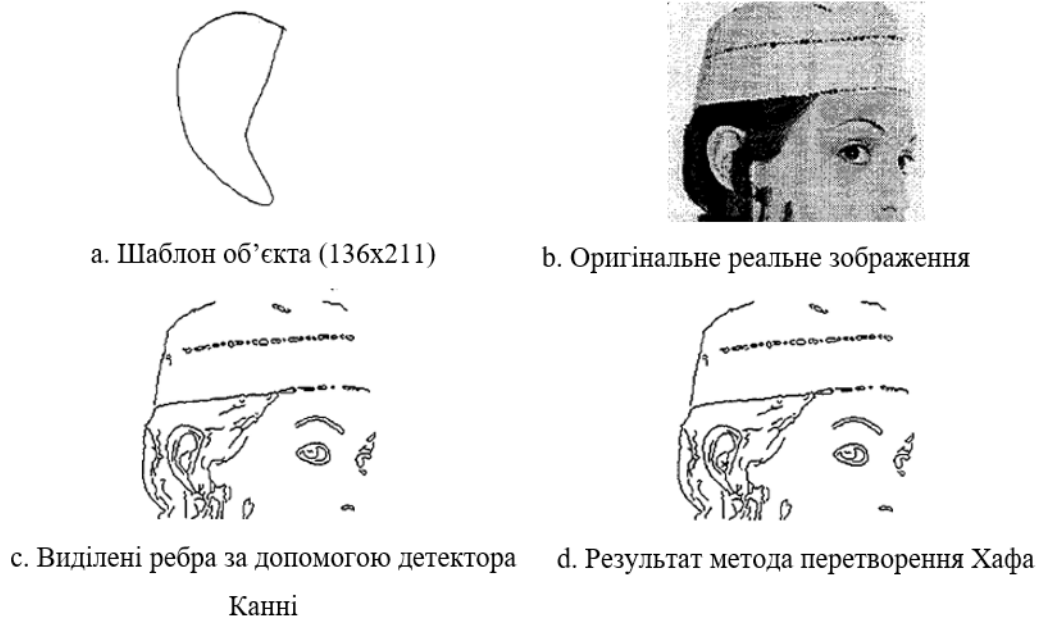


параметрів  $(\hat{m}, \hat{c})$  набір точок зображення. Ми використали символ капелюха для позначення величин в області відображення. Відображення є відображенням один до багатьох з простору можливих значень параметрів у простір точок зображення. Перетворення Хафа використовує ідею, що рівняння (3) можна розглядати як взаємне обмеження між точками зображення і точками параметрів, і тому його можна інтерпретувати як таке, що визначає відображення "один до багатьох" від точки зображення до множини можливих значень параметрів [2, с. 87–116].

Класичне перетворення Хафа розпізнає лінії шляхом гістографування комбінацій параметрів, що відповідають ізольованим локальним спостереженням точок-кандидатів. Основна послідовність кроків для декартового перетворення Хафа може бути виражена наступним чином.

- 0) Представлення зображення в декартових координатах.
- 1) Вибрати точки-кандидати в декартовій області зображення (наприклад, за допомогою виявлення країв).
- 2) Для кожної точки-кандидата намалюйте синусоїду області Хафа, яка відповідає всім прямим, що проходять через цю точку.
- 3) Побудувати гістограму "влучень" в області Хафа, отриману в процесі малювання.
- 4) Виділити області піків, координати яких в області Хафа є полярними координатами вершин розпізнаних ліній.
- 5) Побудувати розпізнані лінії в області зображення, перевівши точки області Хафа з полярних координат в декартові і провівши через них прямі, перпендикулярні до прямих, проведених з них до початку координат [3].

Автори модифікованого метода перетворення Хафа продемонстрували порівняння даного метода з методом детектора границь Канні [4, с. 511–515].



**Рис. 3. Результат виявлення об'єкта на реальному зображенні**

Перетворення Хафа дозволяє виявляти об'єкти навіть у випадках, коли вони перекриваються або знаходяться під різними кутами. Він застосовується у багатьох областях, включаючи медичну діагностику (виявлення кісткових структур на рентгенограмах), комп'ютерне бачення, відеоспостереження (виявлення ліній на дорогах або об'єктів у відео) і багато інших.

Важливо відзначити, що хоча перетворення Хафа є потужним методом, воно може бути обчислювально витратним, особливо при великій кількості точок на зображенні. Також варто враховувати, що точність результатів може залежати від параметрів, вибраних для перетворення.

**Перетворення Гарріса**, також відоме як детектор кутів та країв Гарріса, є важливим інструментом у комп'ютерному зорі та обробці зображень. Воно було розроблене Крісом Гаррісом та Майклом Стівенсом і вперше представлено в 1988 році. Перетворення Гарріса використовується для виявлення кутів та країв на зображеннях, а також для визначення важливих точок, які можуть бути використані для подальших завдань, таких як розпізнавання об'єктів та відстеження руху.



Детектор кутів Гарріса заснований на детекторі кутів Моравека, який працює шляхом аналізу локального вікна на зображенні та визначення середніх змін інтенсивності, які виникають при зсуві цього вікна на невелику відстань в різних напрямках. Цей процес потребує розгляду трьох різних сценаріїв [5, с. 3-4]:

А. Якщо ділянка зображення у вікні пласка (тобто приблизно постійна за інтенсивністю), то всі зсуви призведуть лише до невеликої зміни;

В. Якщо вікно перетинає край, то зсув уздовж краю призведе до невеликої зміни, але зсув перпендикулярно до краю призведе до великої зміни;

С. Якщо ділянка з вікном є кутом або ізольованою точкою, то всі зсуви призведуть до великих змін. Кут можна виявити, знайшовши, коли мінімальна зміна, спричинена будь-яким зсувом, є великою.

Тепер дамо математичну специфікацію вищесказаного. Позначимо інтенсивність зображення через  $I$ , а зміну  $E$ , спричинену зсувом  $(x,y)$ , - через:

$$E_{x,y} = \sum_{u,v} w_{u,v} [I_{x+u,y+v} - I_{u,v}]^2, \quad (4)$$

де  $w$  задає вікно зображення: воно дорівнює одиниці в межах заданої прямокутної області, і нулю в інших місцях. Зсуви  $(x,y)$ , що розглядаються, складаються з  $\{(1,0), (1,1), (0,1), (-1,1)\}$ . Таким чином, детектор кутів Моравека полягає в наступному: шукати локальні максимуми в  $\min\{E\}$  вище деякого порогового значення.

При цьому, оператор Моравека мав низку проблем:

1. Відгук є анізотропним, оскільки розглядається лише дискретний набір зсувів на кожні 45 градусів.
2. Відгук є зашумленим, оскільки вікно є бінарним і прямокутним.
3. Оператор занадто швидко реагує на краї, тому що враховується лише мінімум  $E$ .

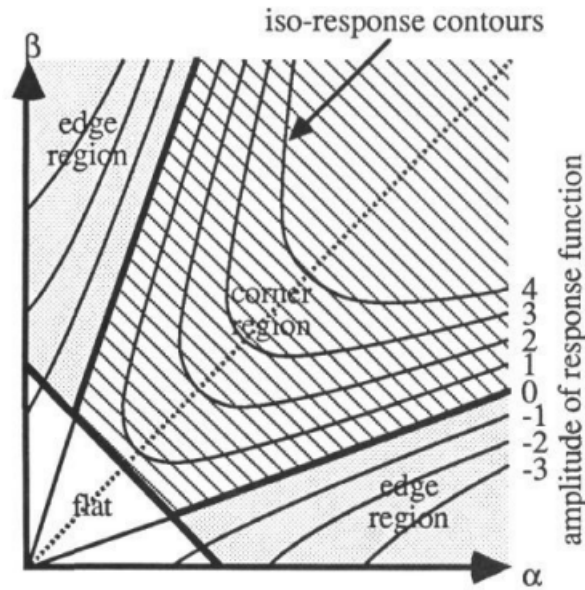
Вирішивши їх, Гарріс вдосконалив цей метод, зробивши його більш точним і надійним. Як і до цього, розглянемо три сценарії:

А. У випадку, коли обидві кривизни малі і локальна автокореляційна функція є плоскою, віконна область зображення має майже постійну інтенсивність (тобто зсуви ділянки зображення мало змінюють  $E$ ).

В. Якщо одна кривизна велика, а інша мала, і локальна автокореляційна функція має форму гребеня, то лише зсуви вздовж гребеня (тобто вздовж краю) призводять до невеликих змін в  $E$ , що вказує на наявність краю.

С. Якщо обидві кривизни великі і локальна автокореляційна функція має гострий пік, то зсуви в будь-якому напрямку призводять до збільшення  $E$ , що вказує на наявність кута.

Ми можемо представити це на графіку простору  $(\alpha, \beta)$ . Ідеальне ребро матиме велике значення  $\alpha$  і нульове значення  $\beta$  (це буде поверхня трансляції), але на практиці  $\beta$  буде досить малим порівняно з  $\alpha$  через шум, пікселізацію та квантування інтенсивності. Кут буде позначатися великими значеннями  $\alpha$  і  $\beta$ , а пласка область зображення - малими значеннями  $\alpha$  і  $\beta$ . Оскільки збільшення контрастності зображення в  $p$  разів збільшує  $\alpha$  і  $\beta$  пропорційно на  $p^2$ , то якщо  $(\alpha, \beta)$  вважається крайовою областю, то і  $(\alpha p^2, \beta p^2)$ . також, для додатних значень  $P$ . Аналогічні міркування застосовуються і до кутів. Таким чином, простір  $(\alpha, \beta)$  слід поділити, як показано товстими лініями на рис. 4 [6].



**Рис. 4.** Автокореляція головної кривизни простору - товсті лінії дають класифікацію кутів/країв/площин, тонкі лінії - контури рівного відгуку

Перетворення Гарріса застосовується в багатьох областях, включаючи відстеження руху в відео, стереозор та реконструкцію 3D-сцен, розпізнавання об'єктів, автоматичну фокусування камер, робототехніку, а також для поліпшення загальної обробки зображень.

Однак важливо враховувати, що ефективність перетворення Гарріса може знижуватися на зображеннях з великим розмахом освітлення або на зображеннях з великою кількістю текстури. У таких випадках інші методи, такі як методи можуть бути більш ефективними.

**Blob Analysis** (також відомий як аналіз зв'язаних компонентів або маркування областей) - це фундаментальний метод машинного зору, заснований на аналізі послідовних областей зображення. Як такий, він є інструментом вибору для додатків, в яких об'єкти, що перевіряються, чітко виділяються на тлі фону. Різноманітний набір методів Blob Analysis дозволяє створювати індивідуальні рішення для широкого спектру завдань візуального контролю[7]. Аналіз згустків широко використовується в обробці зображень і програмах комп'ютерного зору, таких як виявлення об'єктів, відстеження та сегментація зображень.

Основні етапи аналізу плям такі:

1. **Порогове значення:** Першим кроком є застосування операції порогового значення до зображення. Порогове значення перетворює відтінки сірого або кольорове зображення на бінарне зображення, де пікселі вважаються або переднім планом (плямою), або фоном. Цей крок допомагає відокремити об'єкти інтересу від фону.

2. **Маркування з'єднаних компонентів:** Після встановлення порогових значень алгоритм сканує бінарне зображення для виявлення груп з'єднаних пікселів переднього плану. Кожній групі з'єднаних пікселів присвоюється унікальна мітка або ідентифікатор. Цей процес називається маркуванням зв'язаних компонентів, і він зазвичай реалізується за допомогою таких методів, як пошук в глибину або пошук в ширину.

3. **Фільтрація та аналіз згустків:** Після завершення маркування зв'язаних компонентів алгоритм може застосувати фільтрацію для усунення занадто малих або занадто великих згустків, залежно від вимог програми. Фільтрація допомагає видалити шум і нерелевантні області, гарантуючи, що розглядаються лише релевантні об'єкти.

4. **Вилучення характеристик згустків:** Після отримання згустків, які вас цікавлять, можна виділити різні характеристики, щоб охарактеризувати та проаналізувати кожен з них. Найпоширеніші характеристики включають центроїд (центр маси), площу (кількість пікселів), обмежувальну рамку (прямокутник, що оточує згусток), периметр та ексцентриситет (видовження форми). Ці характеристики використовуються для розрізнення різних блобів і розуміння їхніх властивостей.

5. **Відстеження та розпізнавання об'єктів:** Аналіз згустків часто використовується разом з алгоритмами відстеження об'єктів для відстеження руху об'єктів у часі у відеопотоці. Унікальні мітки, призначені

під час маркування підключених компонентів, можна використовувати для відстеження об'єктів від кадру до кадру.



Рис 5. Порогове зображення



Рис 6. Виділення «плями» за допомогою алгоритма

Області застосування Вlob-аналізу включають:

- **Виявлення об'єктів:** Ідентифікація та визначення місцезнаходження об'єктів, що цікавлять на зображенні або відео.
- **Відстеження об'єктів:** Відстеження об'єктів, які рухаються в часі, навіть у складних сценаріях.
- **Сегментація зображення:** Поділ зображення на значущі області для подальшого аналізу.
- **Підрахунок об'єктів:** Підрахунок кількості екземплярів певного об'єкта на зображенні або відео.

Аналіз плям є фундаментальною технікою комп'ютерного зору і відіграє вирішальну роль у багатьох реальних програмах, особливо в тих,

які передбачають ідентифікацію та аналіз об'єктів на зображеннях або відеозаписах, а модифікації цього метода покращують його результат у випадках зі складним фоном і різною варіацією освітлення [8].

**Згорткова нейронна мережа (CNN)** - це клас глибоких штучних нейронних мереж прямого поширення, які використовуються для забезпечення точної роботи в задачах комп'ютерного зору, таких як класифікація та виявлення зображень [9]. CNN схожа на традиційні нейронні мережі, але з більш глибокими шарами. Вона має ваги, зсуви та виходи через нелінійну активацію. Нейрони CNN розташовані в об'ємній формі, наприклад, за висотою, шириною і глибиною.

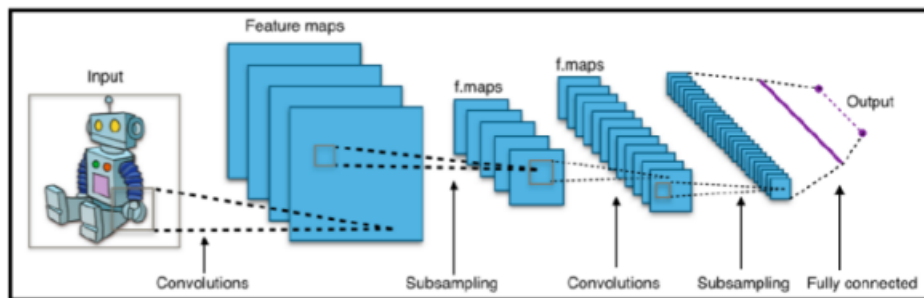


Рис. 7. Архітектура CNN [10]

На рис. 5 показано архітектуру CNN, яка складається зі згорткового шару, об'єднуючого шару та повністю з'єданого шару. Згортковий шар і шар об'єднання зазвичай чергуються, і глибина кожного фільтра збільшується зліва направо, в той час як вихідні розміри (висота і ширина) зменшуються. Повністю з'єднаний шар є останнім етапом, який подібний до останнього шару звичайних нейронних мереж.

Для аналізу зображень було розроблено модифікацію CNN, яка отримала назву "регіони зі згорнутою нейронною мережею" (R-CNN). Цей метод поєднує в собі прямокутні пропозиції регіонів з функціями згорнутої нейронної мережі. R-CNN - це двоетапний алгоритм виявлення. На першому етапі визначається підмножина областей на зображенні, які можуть містити об'єкт. На другому етапі об'єкт класифікується в кожній області.



Моделі виявлення об'єктів за допомогою R-CNN базуються на таких трьох процесах:

- Знайти на зображенні області, які можуть містити об'єкт. Ці регіони називаються пропозиціями регіонів.
- Отримати функції CNN із пропозицій регіону.
- Класифікувати об'єкти за виділеними ознаками.

Є три варіанти R-CNN. Кожен варіант намагається оптимізувати, прискорити або покращити результати одного або кількох із цих процесів.

1. **R-CNN**. Система R-CNN, спочатку, створює пропозиції регіонів використовуючи алгоритм, такий як Edge Boxes. Потім області, що були запропоновані, обрізаються та змінюються у розмірі. Після цього, з використанням нейронної мережі CNN, проводиться класифікація обрізаних та змінених регіонів. Нарешті, машини опорних векторів (SVM), яка навчена на основі функцій CNN, використовуються для подальшого уточнення обмежувальних рамок пропозицій регіонів.

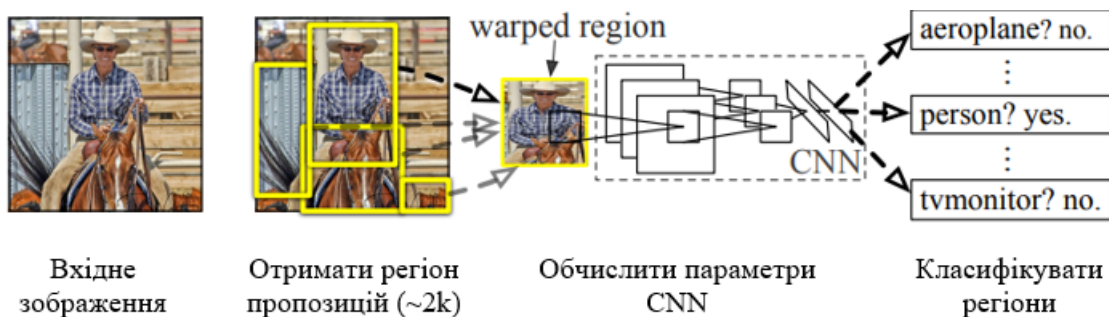


Рис. 8. Архітектура R-CNN [11, с. 1]

2. **Fast R-CNN**. Схоже на детектор R-CNN, детектор Fast R-CNN також використовує алгоритм, відомий як Edge Boxes, для генерації пропозицій регіонів. Але вони відрізняються в способі обробки цих регіонів. У випадку детектора R-CNN, пропозиції регіонів обрізаються та змінюються за розміром перед подачею на вхід мережі. Натомість, детектор Fast R-CNN використовує всю площу зображення для обробки. Також, у відміню від детектора R-CNN, який повинен класифікувати кожен окремий

регіон, Fast R-CNN об'єднує функції згорткової нейронної мережі, які відповідають кожній пропозиції регіону. Основна перевага Fast R-CNN полягає в ефективності, оскільки він виконує обчислення для перекриваючихся областей разом, що робить його більш продуктивним в порівнянні з R-CNN.

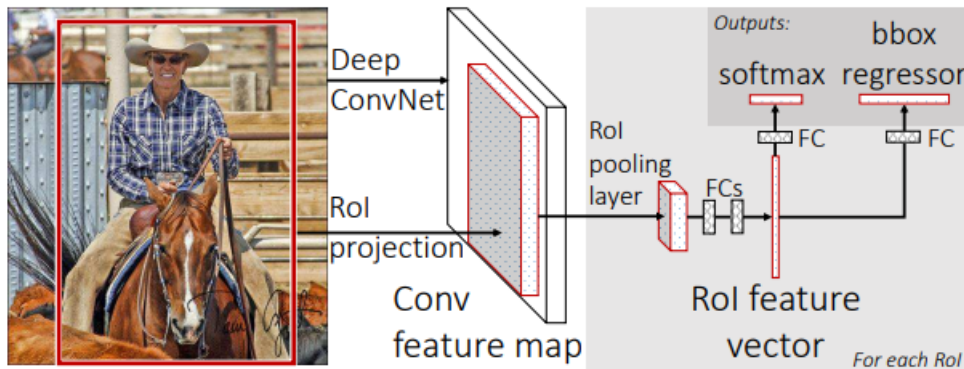


Рис. 9. Архітектура Fast R-CNN [12, с. 1440]

3. **Faster R-CNN.** Детектор Faster R-CNN впроваджує мережу пропозицій регіонів (RPN), яка генерує пропозиції регіонів безпосередньо всередині мережі, у замість використання зовнішнього алгоритму, такого як Edge Boxes. RPN використовує Anchor Boxes для визначення об'єктів. Процес створення регіональних пропозицій у мережі відбувається швидше та має краще налаштовані параметри.

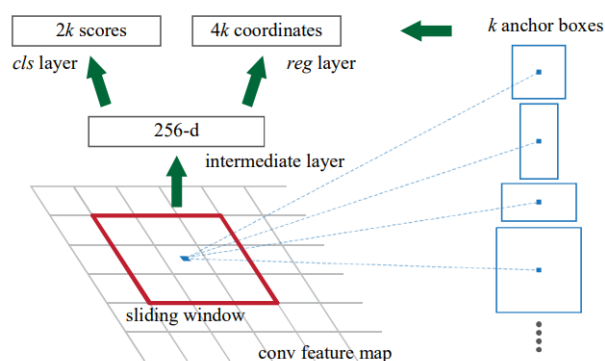


Рис. 10. Архітектура RPN [13, с. 3]

**You only look once (YOLO)** – це згорткова нейронна мережа, яка розглядає виявлення об'єктів як дину регресійну задачу: від пікселів зображення до координат обмежувальних рамок та ймовірностей класів.

Використовуючи цю систему, вам достатньо лише один раз поглянути (you only look once) на зображення, щоб передбачити, які об'єкти присутні і де вони знаходяться.

YOLO напрочуд проста: див. Рисунок N. Одна згорткова мережа одночасно прогнозує кілька обмежувальних рамок і ймовірності класів для цих рамок. YOLO навчається на повних зображеннях і безпосередньо оптимізує ефективність виявлення.

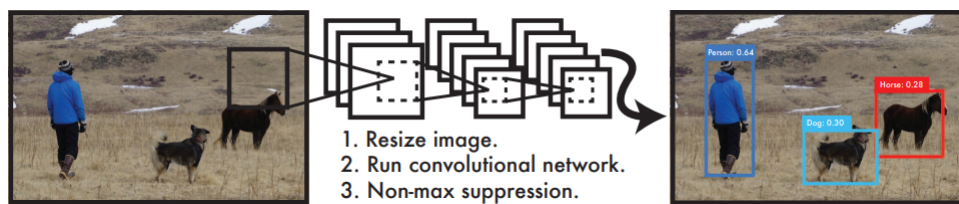


Рис. 11. Архітектура YOLO

YOLO має низку переваг перед іншими методами виявлення об'єктів[14, с. 1-2]:

1. YOLO надзвичайно швидка. Оскільки ця нейронна мережа розглядає виявлення як регресійну задачу, їй не потрібен складний конвеєр.

2. По-друге, YOLO міркує глобально про зображення, коли робить прогнози. На відміну від методів, що базуються на ковзаючому вікні та регіональних пропозиціях, YOLO бачить все зображення під час навчання та тестування, тому він неявно кодує контекстну інформацію про класи та їхній вигляд.

3. YOLO запам'ятовує узагальнені зображення об'єктів. Оскільки YOLO має високий рівень узагальнення, він менш схильний до збоїв при застосуванні до нових областей або несподіваних вхідних даних.

**Результати та висновки.** У цій статті ми провели огляд двох важливих підходів до обробки зображень у галузі комп'ютерного зору: нейронних мереж і стандартних алгоритмів. Вони обидва грають важливу роль у різних аспектах сучасних технологій та застосувань, і мають свої сильні сторони та обмеження.

Нейронні мережі, зокрема з використанням глибокого навчання, стали потужним інструментом у розв'язанні завдань обробки зображень. Вони здатні виявляти складні патерни та особливості у даних, автоматично вчитися з них і адаптуватися до нових завдань. Особливо ефективні вони у завданнях, де великі обсяги даних та варіативність інформації є ключовими факторами.

З іншого боку, стандартні алгоритми обробки зображень все ще залишаються важливими, особливо в сценаріях, де вимоги до швидкості та обчислювальних ресурсів обмежені. Вони можуть бути більш ефективними у виявленні конкретних структур або особливостей на зображенні.

Зрозуміло, що оптимальний вибір між цими підходами залежить від конкретного завдання і ресурсів, які доступні для розв'язання цього завдання. В багатьох випадках комбінування цих двох підходів може бути найкращим варіантом, де нейронні мережі використовуються для виявлення загальних особливостей, а стандартні алгоритми для більш точної локалізації.

Усього враховуючи, розвиток обробки зображень в галузі комп'ютерного зору надзвичайно захоплюючий і продовжує динамічно розвиватися. Подальший внесок обох підходів у цю галузь буде ключовим для досягнення нових висот і розв'язання складних завдань, що стоять перед сучасним суспільством.

### **Література**

1. Canny J. A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI. 1986. 8(6). P. 679-698. doi:10.1109/TPAMI.1986.4767851.
2. Illingworth J., Kittler J. A survey of the hough transform. Computer Vision, Graphics, and Image Processing. October 1988. Vol. 44, Iss. 1. P. 87-116.

3. Hough P. V. C. A method and means for recognizing complex patterns, U.S. Patent 3,069,654.
4. Pina-Fu Fung, Wing-Sze Lee, King I. Randomized generalized Hough transform for 2-D gray scale object detection. *IEEE Proceedings of 13th International Conference on Pattern Recognition*. Vienna, Austria (1996.08.29-1996.08.29). Proceedings of 13th International Conference on Pattern Recognition. 1996. Vol. 2. doi: 10.1109/ICPR.1996.546877
5. Moravec H. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover, Tech Report CMU-RI-TR-3, Carnegie-Mellon University, Robotics Institute. September 1980.
6. Harris C., Stephens M. A combined corner and edge detector, in *Alvey Vision Conference*. Manchester, UK, 1988. Vol. 15.
7. Aurora Vision Documentation. URL: [https://docs.adaptive-vision.com/current/studio/machine\\_vision\\_guide/BlobAnalysis.html](https://docs.adaptive-vision.com/current/studio/machine_vision_guide/BlobAnalysis.html) (дата звернення: 02.09.2023)
8. Jia T., Sun N., Cao M. Moving object detection based on blob analysis. *IEEE International Conference on Automation and Logistics*. 2008.
9. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems*, 2012.
10. Gulli A., Pal S. *Deep Learning with Keras*. Birmingham: Packt, 2017. 318 p.
11. Girshick R., Donahue J., Darrell T., Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014.
12. Girshick R. Fast r-cnn. *Proceedings of the IEEE International Conference on Computer Vision*. 2015.

13. Shaoqing R., He K., Girshick R., Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*. 2015. Vol. 28.
14. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection. May 2016. P. 779-788.