Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА ТА ДОСЛІДЖЕННЯ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ ЗА ДОПОМОГОЮ MASK R-CNN, GRABCUT І OPENCV
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-18-2

Теребецький М. А.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник проф. Кузьомін О.Я.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Теребецькому Микиті Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та дослідження сегментації зображень, за допомогою Mask R-CNN, Grabcut і OpenCV
затверджена наказом університету від 16 травня 2022 року № 541Ст
2. Термін подання студентом роботи до екзаменаційної комісії 23 травня 2022 р.
3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, середовище розробки програмних систем Microsoft Visual Studio та Visual Studio Code, JetBrains Pycharm, бібліотека комп'ютерного зору з відкритим кодом OpenCV.
4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз існуючих методів сегментації зображень.
2. Реалізувати метод обробки, використовуючі мережі Grabcut і Mask R-CNN.
3.Створити користувацький застосунок на основі реалізованого методу.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Приклади та схеми роботи, та зображення архітектур алгоритмів сегментації зображень, .

6. Консультанти розділів роботи (п.6 включається до завдання 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
Консультант з дотримання діючих стандартів та норм	Доцент Бєлова Н.В.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	18.04.2022	
2	Аналіз завдання, підбір літератури	18.04.22-21.04.22	
3	Аналіз літератури з досліджуваної проблеми	22.04.22-25.04.22	
4	Аналіз технічних засобів	26.04.22-30.04.22	
5	Розробка інформаційної системи вибраної предметної області	01.05.22-14.05.22	
6	Програмна реалізація	15.05.22-23.05.22	
7	Оформлення пояснювальної записки	24.05.22-26.05.22	
8	Перевірка на плагіат	27.05.22	
9	Рецензування	28.05.22	
10	Підготовка презентації та доповіді	29.05.22-30.05.22	
11	Занесення роботи в електронний архів	31.05.22	
12	Попередній захист кваліфікаційної роботи	31.05.22	

Дата видачі завдання 18 квітня 2022 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Кузьомін О.Я.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 44 с., 26 рис., 2 дод., 34 джерела.

АЛГОРИТМ, K-СЕРЕДНІХ, СЕГМЕНТАЦІЯ ЗОБРАЖЕНЬ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА.

Об’єктом роботи є набір зображеннь, із різноманітними об’єктами, в тому числі і перекриваючими друг друга, та відмінним контрастом із заднім планом.

Метою роботи є розробка застосунка, що базуються на використанні технологій Mask R-CNN Grabcut і OpenCV для поділення зображення на окремі об’єкти.

Дані технології використовують алгоритм к-середніх для навчання моделі кольорів та алгоритмі GraphCuts безпосередньо для сегментації.

У результаті роботи здійснена розробка програмного забезпечення для розпізнавання об’єктів на зображенні.

ALGORITHM, K-MEANS, IMAGE SEGMENTATION, CONVOLUTIONAL NEURAL NETWORK.

The object of the work is the set of images with different objects, including those that overlap, and various contrast with background.

The aim of the work is to develop application, which based on the usage Mask R-CNN, Grabcut and OpenCV technologies to divide image into objects.

Used methods based on k-middle algorithm for color model learning and GraphCuts directly for segmentation.

As a result of developing was created software implementation of the image object discover system .

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЗНМ – згорткова нейронна мережа

RGB – кольорова модель за ознаками наявності червого, зеленого та синього кольорів.

HSI – кольорова модель за ознаками тону, насиченості та світлоти

CNN – Convolutional Neural Network\

OpenCV – Бібліотека компьютерного зору, обробки зображень, так загальних чиселбних алгоритмів з відкритим вихідним кодом.

ВСТУП

Коли людина розглядає будь-яку картинку, вона здатна інстинктивно розділяти її на частини, наприклад розглядаючи своє фото, вона може виділити своє обличчя, але комп’ютеру для цього потрібно задати певні інструкції, їх набір і називається сегментацією зображення.

Основна мета сегментації зображень, поділити їх на частини, за якими-небудь критеріями чи атрибутами, що дасть змогу класифікувати окремі пікселі, та розглядати певні їх групи для подальшої обробки, наприклад для аналізу системою ідентифікування особи і подальшої авторизації у деякій системі.

Актуальність роботи полягає у тому що в нас час набуває необхідності використання систем комп’ютерного зору для автоматизації або для узручнення у багатьох сферах повсякденного життя. Такі системи у переважній більшості потребують алгоритми, що розділяють зображення на певні частини для коректного аналізу та обробки в інших модулях цих застосунків.

Такі системи широко використовуються у багатьох сферах [33]:

– ідентифікація номерних знаків, – контроль за дорожнім рухом, у нас час здійснюється за допомогою камер, в яких алгоритми сегментації, допомагають з’ясувати, чи було порушено правила дорожнього руху, якщо так то ким саме, та автоматично надіслати йому квитанцію, для погашення штрафу;

– розпізнавання обличчя, – алгоритм що створений для авторизації користувача у деякій системі, він за допомогою камери фотографує обличчя, та перевіряє чи відповідає воно збереженому зразку, щоб надати користувачу доступ до цієї системи, компанія Apple впровадила сервіс, що використовує таку технологію, у своїй операційній системі IOS, він називається FaceID;

– пошук по зображенню, – останнім часом популярності набули сервіси, що дають змогу знайти певну інформацію у мережі інтернет, із вхідним

набором даних, у вигляді зображення. Такі системи теж сегментують зображення, щоб результати пошуку стосувалися не лише картинки в цілому, а і деяких частин, які можуть цікавити користувача;

– медична візуалізація, – у медицині, все більше стає необхідним знаходити сторонні тканини чи клітини, що дасть змогу ідентифікувати деяку хворобу, або пухлину, щоб вчасно почати лікування. У наш час сегментація зображень це вже не просто можливість відредагувати зображення, а це повноцінний інструмент, завдяки якому можливо навіть моделювати хірургічні операції, із здійсненням контролю та керуванням.

Хоча було згадано безліч глобальних областей, але існує і багато інших галузей, в яких теж частково задіяні згадані алгоритми, так як охоронні системи, сіль. господарство, виробництво, та ін. Із часом, значення таких алгоритмів лише зростає, а список галузей в яких їх слід використовувати розширюється.

1 ОГЛЯД ОСНОВНИХ МЕТОДІВ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ

1.1 Сегментація за пороговим значенням

Найпростішим алгоритмом сегментації зображень є метод, що базується на встановленні певного порогового значення кольору пікселя такого, щоб розділи усі точки зображення на два класи, ті пікселі значення яких менше за порогове позначаются нулем, ті в яких більше одиницею, таким чином відбувається бінаризація зображення і воно перетворюється на двійкову мапу. Такий метод ефективний у випадках коли різниця значень пікселів між классами досить велика, і можна визначити коректне порогове значення. Також слід зазначити, що є два підходи то цієї технології, глобальна обробка – коли обробляється усе зображення, локальна – лише його частина. Найполярнішими методами що викроистовують цю технологію є:

- метод Отса (Локальна обробка);
- метод Бернса (Глобальна обробка);
- метод Ейквілу (Глобальна обробка);
- метод Ніблека (Глобальна обробка);
- метод Яновиця та Брукштейна (Глобальна обробка).

На рисунку 1.1 наведено приклад роботи сегментації за таким алгоритмом.

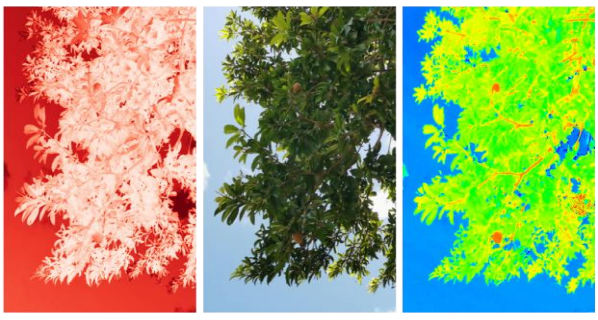

Рисунок 1.1 - Приклад сегментації за пороговим значенням [23]

Лістинг 1.1 Реалізація сегментації за пороговим значенням

```
import numpy as np
import glob
import matplotlib.pyplot as plt
import skimage.io
import skimage.color
import skimage.filters
%matplotlib widget

image = skimage.io.imread("data/shapes-01.jpg")
fig, ax = plt.subplots()
plt.imshow(image)
plt.show()
gray_image = skimage.color.rgb2gray(image)
blurred_image = skimage.filters.gaussian(gray_image, sigma=1.0)
fig, ax = plt.subplots()
plt.imshow(blurred_image, cmap="gray")
plt.show()
histogram, bin_edges = np.histogram(blurred_image, bins=256, range=(0.0, 1.0))
fig, ax = plt.subplots()
plt.plot(bin_edges[0:-1], histogram)
plt.title("Grayscale Histogram")
plt.xlabel("grayscale value")
plt.ylabel("pixels")
plt.xlim(0, 1.0)
plt.show()
t = 0.8
binary_mask = blurred_image < t
fig, ax = plt.subplots()
plt.imshow(binary_mask, cmap="gray")
plt.show()
```

1.2 Сегментація на основі регіонів

Цей метод працює за принципом групування сусідніх пікселів за схожістю, створюючі відповідні класи. Тобто, спочатку обираються початкові пікселі – насіння, потім поступово обходяться інші і приєднуються до відповідних класів, доти, доки не буде сегментоване усе зображення. Критерії якими керуються метод додаючи поточний піксель до певного регіону, це:

– відстань до центру регіону;
– відстань до точки що була включена до класу останньою;
– значення найкоротшої путі від точки до центру.

Завдяки простим розрахункам, і високій швидкості роботи цей метод дуже ефективний, але лише в тих ситуаціях коли об’єкт помітно контрастує на фоні, інакше результат може бути некоректним. Результат обробки зображення задопомогою наведеного алгоритму ми можемо побачити на рисунку 1.2.

Рисунок 1.2 - Приклад сегментації на основі регіонів [24]

Лістинг 1.2 Реалізація сегментації на основі регіонів

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
from skimage.filters import sobel

elevation_map = sobel(coins)
fig, ax = plt.subplots(figsize=(4, 3))
ax.imshow(elevation_map, cmap=plt.cm.gray, interpolation='nearest')
ax.axis('off')
ax.set_title('elevation_map')
markers = np.zeros_like(coins)
markers[coins < 30] = 1
markers[coins > 150] = 2
fig, ax = plt.subplots(figsize=(4, 3))
ax.imshow(markers, cmap=plt.cm.spectral, interpolation='nearest')
ax.axis('off')
ax.set_title('markers')
segmentation = morphology.watershed(elevation_map, markers)
fig, ax = plt.subplots(figsize=(4, 3))
ax.imshow(segmentation, cmap=plt.cm.gray, interpolation='nearest')
ax.axis('off')
ax.set_title('segmentation')
from skimage.color import label2rgb
segmentation = ndi.binary_fill_holes(segmentation - 1)
labeled_coins, _ = ndi.label(segmentation)
image_label_overlay = label2rgb(labeled_coins, image=coins)
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(6, 3), sharex=True, sharey=True)
ax1.imshow(coins, cmap=plt.cm.gray, interpolation='nearest')
ax1.contour(segmentation, [0.5], linewidths=1.2, colors='y')
ax1.axis('off')
ax1.set_adjustable('box-forced')
ax2.imshow(image_label_overlay, interpolation='nearest')
ax2.axis('off')
ax2.set_adjustable('box-forced')
fig.subplots_adjust(**margins)
```

1.3 Сегментація по краях

На відміну від попереднього методу, коли ми говоримо про сегментацію по краях, то маємо на увазі виявлення меж об’єктів за допомогою фільтрів та згорток, завдяки різному значенню граничних пікселів у сірому відтінку. Раніше згадані фільтри зазвичай розраховуються алгоритмами які оцініють градієнт зображення у просторовій площині. Нище наведений приклад сегментації по краям за допомогою наступних алгоритмів:

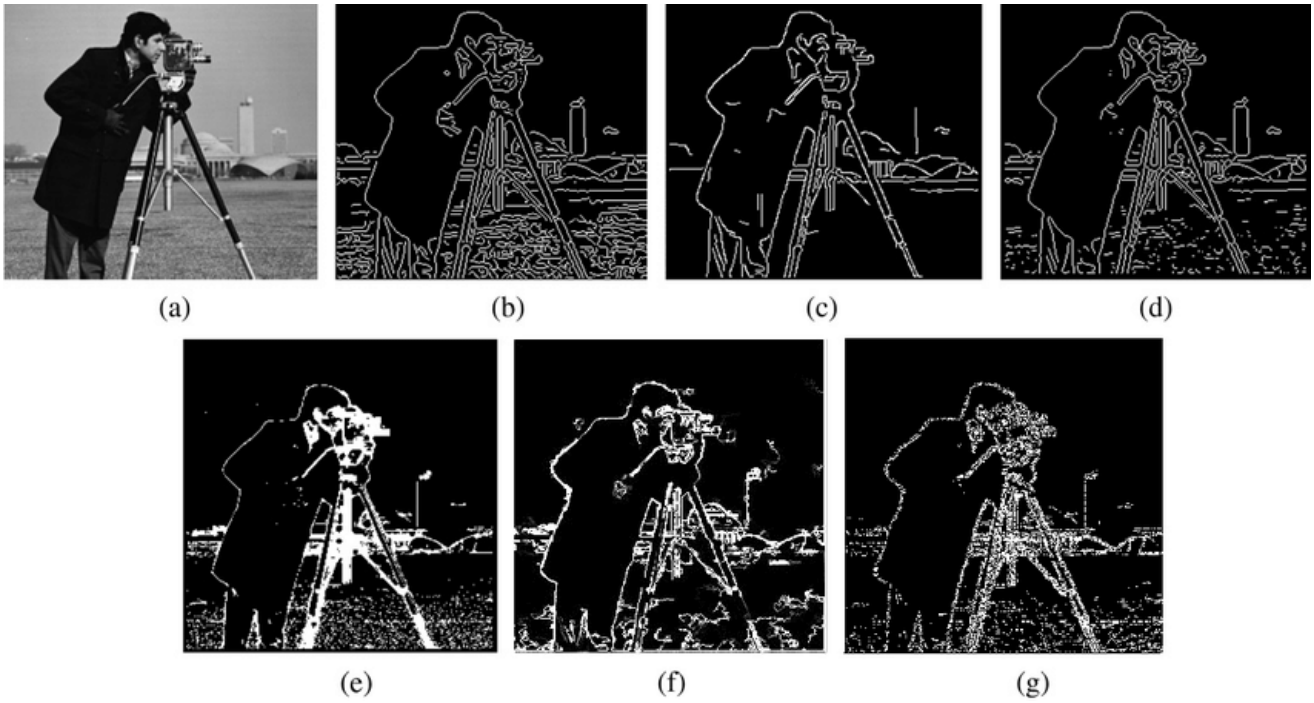

Рисунок 1.3 - Приклад сегментації по краях, за допомогою різних алгоритмів: (a) – Початкове зображення, (b) – Алгоритм Canny, (c) – Алгоритм Edison, (d) – Алгоритм Rothwell, (e) – Алгоритм SUSAN, (f) – Алгоритм виявлення країв на основі теорії всесвітнього тяжіння та оптимізації наслідуванням колонії мурах (Поріг: звичайний), (g) – Очікуваний результат [25]

Цей метод слід використовувати коли на зображенні мало об’єктів і вони достатньо контрастують між собою. Нище наведено приклад його можливої реалізації за допомогою мови Python:

Лістинг 1.3 Приклад реалізації сегментації по краях за допомогою алгоритма Canny

```
import numpy as np
import matplotlib.pyplot as plt
from skimage import data
coins = data.coins()
hist = np.histogram(coins, bins=np.arange(0, 256))
fig, (ax1) = plt.subplots()
ax1.imshow(coins, cmap=plt.cm.gray,interpolation='nearest')
from skimage.feature import canny
edges = canny(coins/255.)
fig, ax = plt.subplots(figsize=(4, 3))
ax.imshow(edges, cmap=plt.cm.gray, interpolation='nearest')
ax.axis('off')
ax.set_title('Canny detector')
Text(0.5, 1.0, 'Canny detector')
```

1.4 Сегментація за допомогою кластеризації

У наш час, набуває необхідності розробка все більше швидких та ефективних методів для обробки зображень. Тому було розроблено метод сегментації за допомогою кластеризації. Один із найрозповсюдженіших алгоритмів такої сегментації, це k-середніх. Спочатку обирається де яка кількість кластерів(відповідно до кількості об’єктів, які потрібно отримати), потім довільно розподіляємо усі точки між кластерами, далі знаходимо центри цих скупчень, після цього обчислюємо відстань від кожної точки до центру до кожного кластера, і перерозподілюємо точки між кластерами за критерієм близькості до центру, повторюємо останні три кроки до тих пір, коли центри

класів співпадатимуть із попередніми, також слід зазначити максимальну кількість ітерацій, адже в деяких випадках при великій кількості даних, деякі пікселі по краях будуть переміщуватися до сусідніх кластерів і збігу їх центрів не відбудеться. На рисунку 1.4 наведене тестове зображення, на 1.5 ми можемо побачити результат розбиття зображення на 3 кластери, на 1.6 результат розбиття на 5 кластерів.

Рисунок 1.4 - Початкове зображення [26]

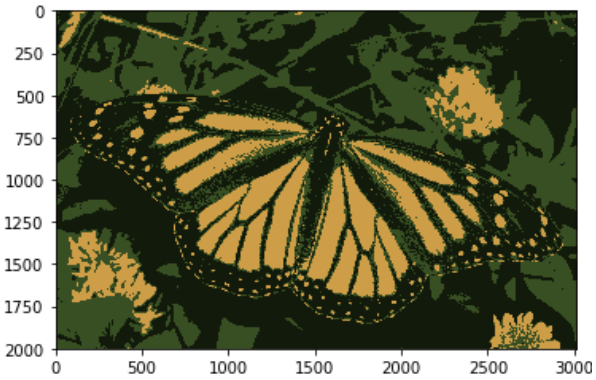

Рисунок 1.5 - Сегментація за допомогою кластеризації алгоритмом k-середніх (3 кластери) [26]

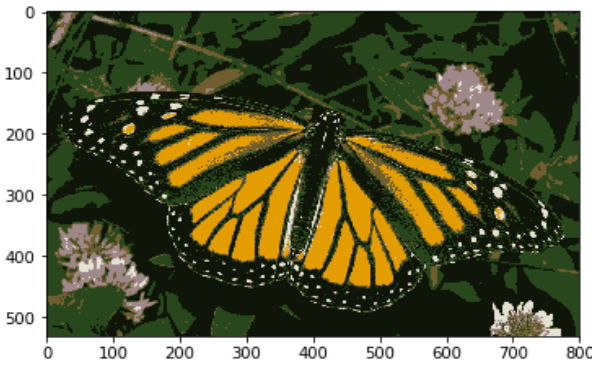

Рисунок 1.6 - Сегментація за допомогою кластеризації алгоритмом k-середніх (5 кластерів) [26]

Лістинг 1.4 Приклад реалізації сегментації за допомогою кластеризації із використання методу *k*-середніх(*k*-means)

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
%matplotlib inline
image = cv2.imread('images/monarch.jpg')
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image)
pixel_vals = image.reshape((-1,3))
pixel_vals = np.float32(pixel_vals)
accuracy)
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100,
0.85)
k = 3
retval, labels, centers = cv2.kmeans(pixel_vals, k, None, criteria, 10,
cv2.KMEANS_RANDOM_CENTERS)
centers = np.uint8(centers)
```

```
segmented_data = centers[labels.flatten()]
segmented_image = segmented_data.reshape((image.shape))
plt.imshow(segmented_image)
```

1.5 Постановка задачі

Таким чином, сегментація зображень є досить важливим кроком для покращення ефективності роботи багатьох систем і в загальному є чи не найвпливовішим чинником автоматизації у самих різних напрямках сучасних технологій. Тому постає завдання розробки застосунку, який міг би використовуючи згорточну нейронну мержу, ефективно виявляти об’єкти на будь-яких зображеннях.

Об’єктом роботи є набір зображеннь, із різноманітними об’єктами, в тому числі і перекриваючими друг друга, та контрастом із заднім планом.

Метою роботи є розробка застосунка, що базується на використанні технологій Mask R-CNN Grabcut і OpenCV для поділення зображення на окремі об’єкти.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз існуючих методів сегментацій зображень;
- реалізувати метод обробки використовуючі мережі Mask R-CNN і Grabcut;
- створити користувацький застосунок на основі реалізованого методу.

2 АЛГОРИТМИ ЗНМ

У згорткових мережах використовується нейрони які мають навчальну вагу та упередження, такі мережі широко використовуються у комп’ютерному зорі, в процесі обробки зображення. В подібних мережах використовуються фільтри, в якості яких виступає частковий вхідний сигнал, це дозволяє системі по-частинно аналізувати і обробляти зображення. Зображення перш за все із кольорової моделі (RGB або HSI) перетворюється у градацію сірого. Відслідкувавши зміни у значеннях пікселів, можна виявити краї зображень, і відповідно класифікувати їх. На рисунку 2.1 зображений приклад роботи подібної мережі.

Рисунок 2.1 - Приклад семантичної сегментації за допомогою ЗНМ [27]

2.1 R-CNN - ЗНМ на основі регіонів

Архітектуру R-CNN (Region-Based Convolutional Neural Network) було розроблено у 2014 році Россом Гіршиком. За допомогою метода selective search ця мережа, знаходить об’єкти і ділить їх на регіони[32]. Вилучає ознаки кожного із отриманих регіонів, за допомогою сгорточних мереж, та класифікує

їх за методом опорних векторів, уточнюючі границі за допомогою лінійної регресії, схему її роботи наведено на рисунку 2.2. На рисунку 2.3 зображено приклад обробки найпростішим алгоритмом із використанням нейронної мережі. Хоча така мережа здатна із високою точністю отримувати регіони з об’єктами та їх класами, але вона потребує великої кількості часу на навчання, і оскільки самі алгоритми поєднання регіонів неефективні, ця мережа не здатна обробляти відео. І Selective search взагалі не является алгоритмом машинного навчання, тому на деяких зображеннях виникають проблеми із виявленням об’єктів. Оскільки ця модель устаріла, зараз вона майже не використовується.

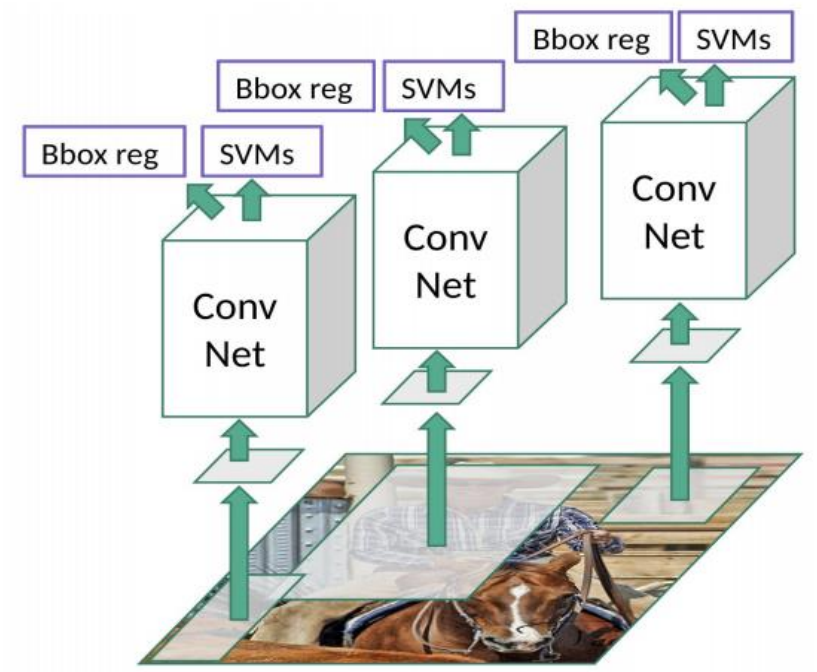

Рисунок 2.2 - Сегментація на основі регіонів із використанням ЗНМ [32]

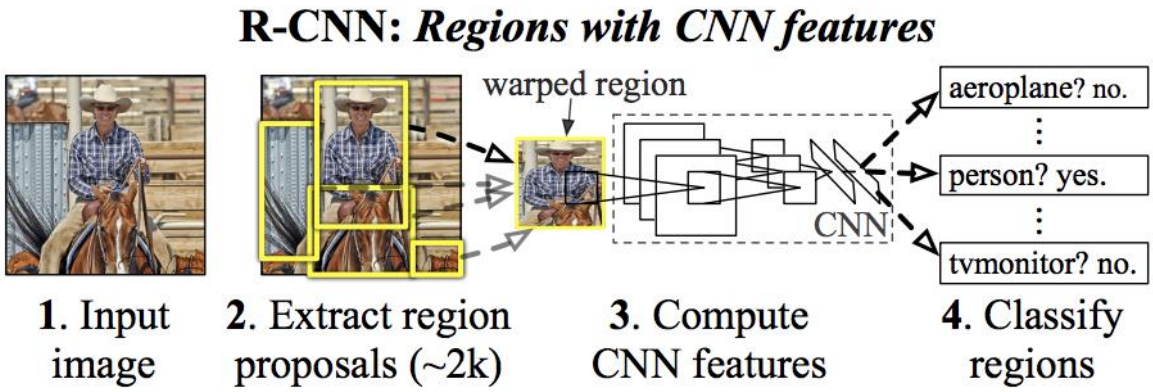

Рисунок 2.3 - Схема роботи R-CNN [28]

2.2 FCN - повністю ЗНМ

Вперше FCN були створені Джонатаном Лонгом, який описав їх у своїй роботі: «Повністю згорткові мережі для семантичної сегментації» [29]. Вона класифікує зображення на рівні пікселів, вирішуючи проблему сегментації на семантичному рівні. На Відміну від звичайної CNN, вона отримує вектори ознак фіксованої довжини використовуючи повністю зв'язаний рівень після згорткового(повний рівень сполучення + вихід softmax). FCN здатна приймати вхідні зображення будь-якого розміру і використовувати рівень деконволюції для останнього рівня згортки. Також ця мережа використовує збільшення дискретизації карти об'єктів, щоб вона поверталася до початкового розміру вхідного зображення, для генерації прогнозу для кожного пікселя, зберігаючи просторову інформацію оброблюваного зображення, щоб перейти на карту об'єктів із збільшеною дискретизацією. На рисунку 2.4 зображено принцип дії архітектури FCN.

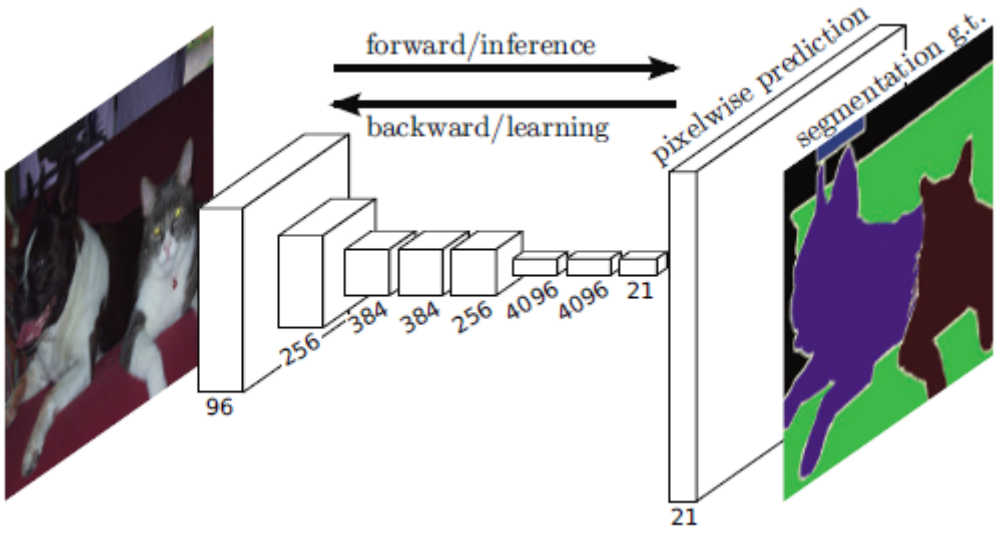

Рисунок 2.4 - візуалізація архітектури FCN [29]

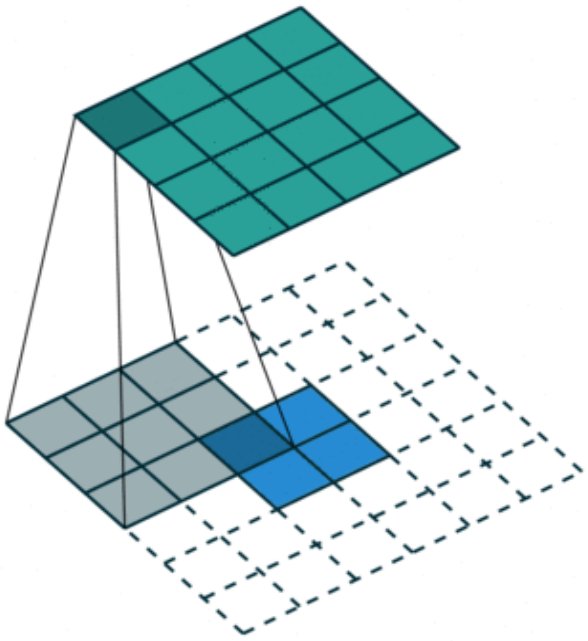

Рисунок 2.5 - Приклад створення згортки для збільшення дискретизації [29]

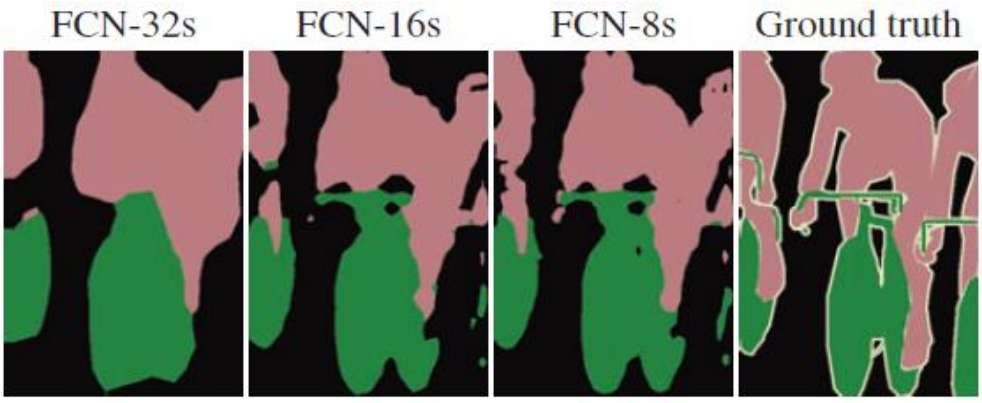

Рисунок 2.6 - Приклад сегментація за допомогою FCN [29]

2.3 U-NET

Слід згадати ще один алгоритм згорткової нейронної мережі, який було розроблено 2017 року у Німеччині, спеціально для сегментації біомедичних зображень. Було використано архітектуру ЗНМ та модернізовано, таким чином, щоб було достатньо меншої кількості зображень для навчання, і можна було отримати більш точні результати. Цей алгоритм здатний обробити зображення, роздільною здатністю 512x512 пікселів, менше ніж за секунду, за умови використання найсучаснішого обладнання [34]. Основним покращенням у архітектурі, стало те, що було доповнено будову шарового механізму, а саме операції пулінгу, тобто зменшення розмірності шарів було замінено операторами збільшення розмірності. Оскільки при таких умовах збільшиться роздільна здатність виходу, наступний згортковий шар здатний навчитися видавати точний результат. Нововведенням стало збільшення каналів ознак у шарах збільшенням розмірності, що надало змогу мережі поширювати контекстну інформацію на шари із більшою роздільною здатністю. Тож структура системи завдяки приблизно симетричному звуженню та розширенню у шарах на буваю U-подібного вигляду. Така мережа використовує лише згортки, минаючи повнозв язні шари. Вона додатково опрацьовує зображення,

щоб прогнозувати значення деяких пікселів і не обмежувати свою роздільну здатність пам’яттю GPU. На рисунку 12 зображено схему роботи цього алгоритму

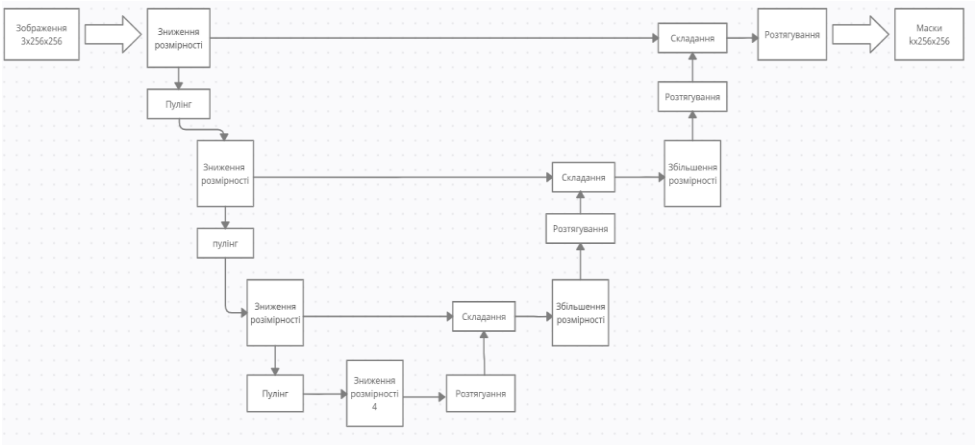

Рисунок 2.7 - Схема роботи U-net

2.4 Fast R-CNN

Оскільки алгоритм описаний у попередньому пункті виявився неефективним, авторам знадобилося покращити його. Так у 2015 році було розробено Fast R-CNN [28]. Його архітектура, побудована на раніше згаданому selective search, але тепер, координати об’єктів конвертуються у координати на мапі ознак. Ця мапа передається на шар RoI(Region of Interest) pooling layer. На кожний окремий регіон, накладається сітка деяким розміром HxW. Для зменшення розмірності застосовується Max Pooling. В такому випадку усі регіони з об’єктами мають однаковий фіксований розмір. Отримані ознаки подаються на повнозв’язний шар(Fully-conectedlayer), який далі розповсюджується на два інших таких же шари. Перший з’ясовує ймовірність належності до певного класу, а інший границі регіону деякого об’єкта. Цей алгоритм показує більш високу точність і та меншу часову затрату, оскільки тепер немає необхідності подавати усі регіони на згортковий шар. Оскільки все

ще використовується Selective Search, у майбутньому алгоритм буде допрацьовано.

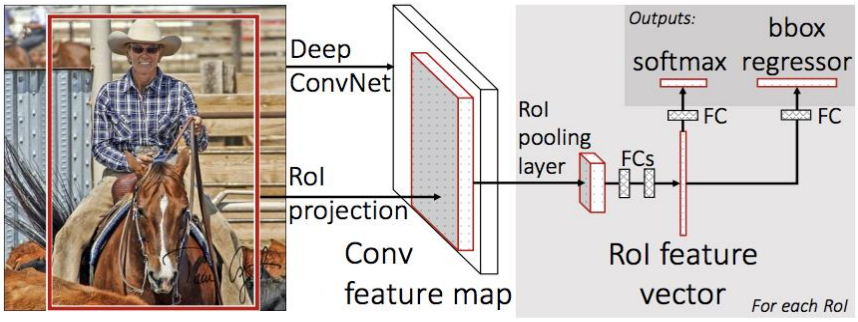

Рисунок 2.8 - Архітектура Fast R-CNN [28]

2.5 Faster R-CNN - Швидша ЗНМ

У 2016 році було продовжено покращення згорткової мережі, і користувачам було запропоновано Faster R-CNN [30]. Тепер замість застарілого і неефективного selective search було розроблено власний метод локалізації - RPN(Region Proposal Networks), в основі якого лежить система якорів.

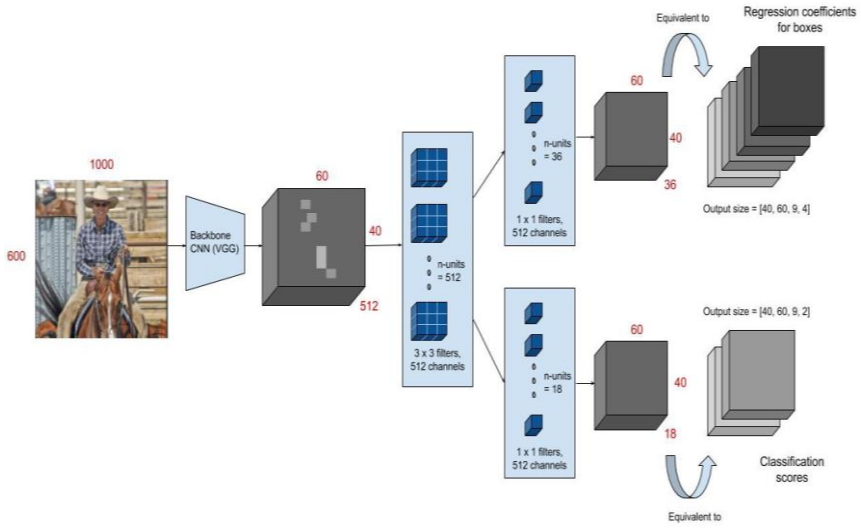

Рисунок 2.9 Архітектура RPN [28]

Мережа працює за наступним алгоритмом: спочатку зображення подається на вхід до ЗНМ, і вона формує мапу ознак. Шар RPN обробляє цю мапу. Ковзаюче вікно(ядро) проходить по усій мапі, центр якого пов’язаний із центрами якорів. Якорі це області, що мають різні співвідношення сторін та різні розміри. Розробники використовують по 3 стандарти для співвідношень та розмірів. За допомогою метрики IoF(intersection-over-union), тобто аналізу степеня перетинання якорів та істинних прямокутників, виноситься рішення про наявність об’єкта у цьому регіоні. Далі за алгоритмом FastCNN також береться мапа ознак і надається слою RoI, з наступною обробкою та класифікацією. Така модель дещо гірше із локалізацією, але працює швидше за попередній алгоритм. Зараз не потрібно вивчати всі архітектури, адже існують вже навчені моделі так як tensorflow models.

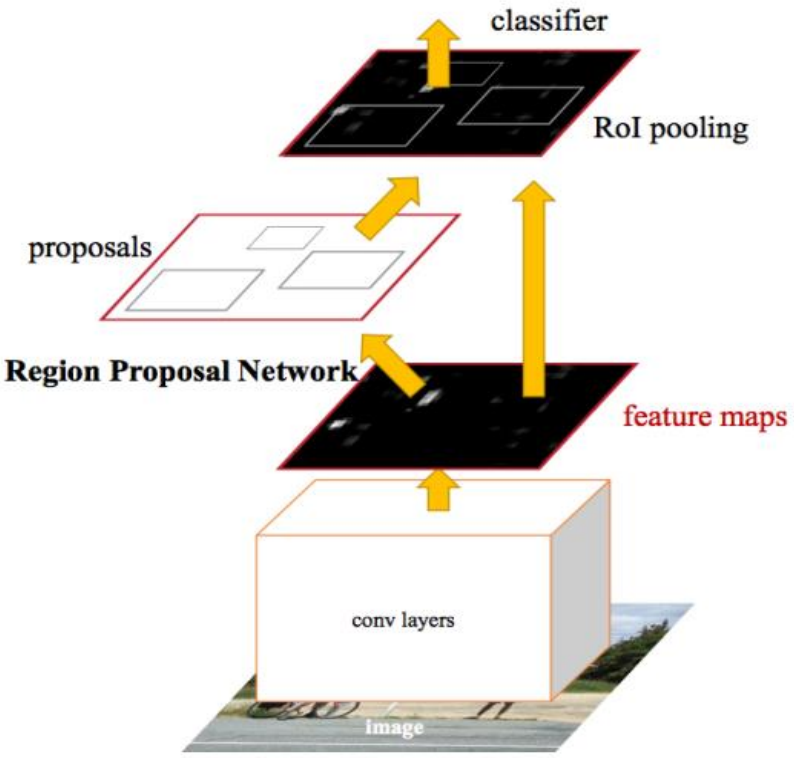

Рисунок 2.10 - Архітектура Faster R-CNN [28]

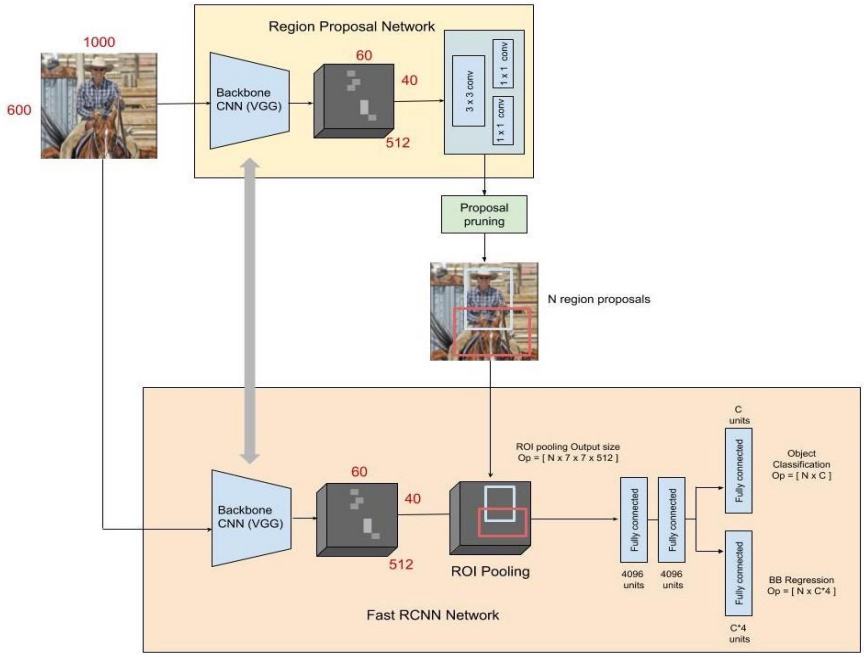

Рисунок 2.11 - Алгоритм пошуку зображення за методом Fast R-CNN в поєднанні з RPN [30]

2.6 Mask R-CNN

У 2017 році було розроблено новий революційний алгоритм сегментації, за основу був взятий Faster-CNN розроблений двома роками раніше, й удосконаленний групою розробників Facebook Research [31]. Мережа з такою архітектурою здатна виділяти контури різних об’єктів, навіть якщо їх безліч, вони різного розміру, та частково перекриваються. Також така мережа може визначати пози людини на фото. Використовуємий алгоритм побудований на двох основних принципах: Convolution, Max Pooling. З Згорточний шар дозволяє об’єднувати значення сусідніх пікселів і виявляти більш загальні ознаки зображення. Для цього по поверхні полотна ковзають квадратнім вікном деякого не великого розміру(приклад: 5x5, 7x7). Таке вікно називається ядром, а кожен його елемент, має свою вагу яку домножають на значення пікселя по

якому воно зараз проходить. Далі складаємо отримані результати щоб отримати значення чергової ознаки. Поступово просуваючи ядро по вертикалі та горизонталі отримуємо мапу ознак усього зображення. У наступних шарах згортка застосовується до цих матриць. У рамках кожного шару, ці мапи можуть скануватися декількома незалежними фільтрами, і на виході відповідно давати декілька матриць. На відміну від попередньої технології Max-Pooling зазвичай використовують для неперетинаючихся групп пікселів. Принцип його роботи це виділити один піксель із кожної групи по заданному правилу, наприклад той що має найбільше значення

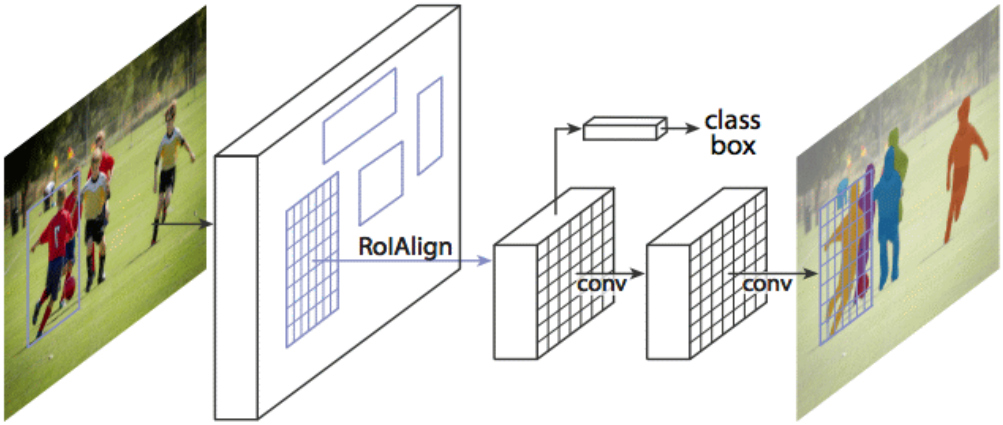

Рисунок 2.12 - Архітектура Mask R-CNN [31]

3 РОЗРОБЛЕННЯ СИСТЕМИ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ НА ОСНОВІ ЗНМ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ GRUBCUT І MASK R-CNN OPENCV

3.1 Підготовка до розробки

Перш ніж переходити безпосередньо до розробки системи сегментації зображень, нам потрібно обрати підхід до її реалізації, тобто алгоритми, методи які ми будемо використовувати в ході створення застонку.
Отже спочатку розпишемо структуру майбутнього проекту:

3.1.1 Головна сторінка

Після запуску додатку користувач, має бачити інтерфейс, до якого входить панель керування, і основне вікно, в якому знаходитиметься полотно для огляду зображення.

3.1.2 Панель керування

Панель повинна містити інструменти: для вибору та зберігання зображення, також для вибору методу обробки зображення, а також маркер для обрання маски (для деяких фільтрів).

3.1.3 Алгоритми сегментації

Наступним, слід обрати алгоритми та методи, за якими буде реалізлвано застосунок. Оскількі маємо за мету створити додаток який сегментуватиме за допомогою Mask R-CNN та Grabcut, тож і використаємо ці мережі. Для цього можна скористуватися публічною бібілотекою OpenCV яка призначена для використання саме вище згаданих технологій.

3.1.4 Інші матеріали

Оскільки ми будемо використовувати готову мережу нам не знадобиться навчання, а для перевірки на якість та коректність роботи ми використаємо ряд заздалегідь підготовлених зображень, які будуть наведені в додатку.

3.2 Огляд програмних засобів для реалізації системи сегментації зображень

Для створення програмної реалізації застоснку сегментації зображень, було використано:

- осередки: Microsoft Visual Studio, Visual Studio Code, PyCharm;
- інші додатки: Google Docs, Google Drive, AWS(Amazon Web Service), Microsoft Office(Word, PowerPoint);
- Мова C#(.Net), – для створення обгортки, тобто робочої поверхні;
- Мова Python, – для безпосереднього використання бібліотек що містять алгоритми сегментації.

Бібліотеки(C#):

– IronPython, – Інтеграція пайтону у C#

Бібліотеки(Python):

– NumPy, Основний модуль для роботи із математичними операціями;
– ArgParse, Модуль обробки аргументів командної строки;
– CV2, Бібліотека алгоритмів OpenCV;
– ImUtils, Пакет з додатковими функціями для спрощення використання OpenCV;
– OS, Модуль взаємодії із операційною системою;
– Time, Робота з часом.

Рішення використати саме вказані засоби, було обумовлено тим, що в цих IDE було організовано зручну та ефективну розробку власних додатків. Ці програми надають можливість використовувати платформу .Net у повній мірі, оскільки їх розробник є автором цього стандарту програмування та мови C#, яку він використовує. Операційна система Windows також є продуктом цього розробника, саме тому ця платформа дозволяє легко і докладно працювати з вікнами та графікою у цій системі. Також це середовище підтримує і Python-розробку, що надає можливість працювати із усіми модулями застосунку у одному вікні. Інтерфейс цієї IDE наведно на риснуку 3.1:

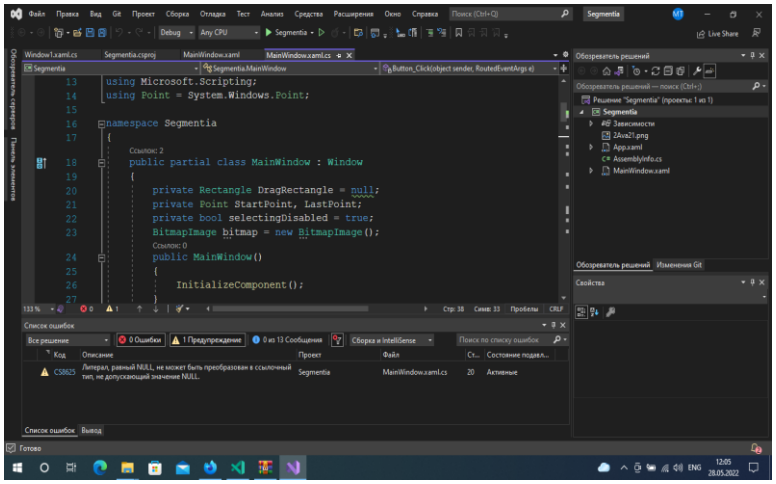

Рисунок 3.1 – Приклад інтерфейсу Microsoft Visual Studio

3.3 Структура проекту

Отже готовий додаток складається із двух частин, користувацького інтерфейсу та інтегрованої за допомогою python-скрипту нейронної мережі, нище наведено її структуру (рис 3.2):

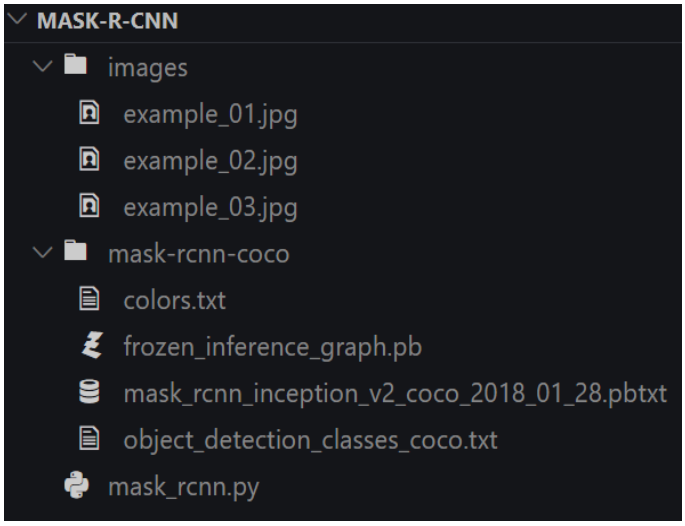

Рисунок 3.2 – Структура проекту

- images, папка із тестовими зображеннями;
- colors.txt, перелік значень кольорів які може використати мережа для позначення об’єктів на зображеннях;
- frozen_inference_graph.pb, заздалегідь навчена вагова модель;
- mask_rcnn_inception_v2_coco_2018_01_28.pbtxt, файл конфігурації моделі;
- object_detection_classes_coco.txt, перелік категорій відомих цій моделі;
- mask_rcnn.py, діючий скрипт, що обробляє зображення за попередньо обраним алгоритмом.

3.4 Інструкція користувача

Після запуску застосунку, буде відображено початкове вікно (рис. 3.3). Натиснувши кнопку Choose Image буде відкрито діалогове вікно для вибору файлу з диску. Скориставшись цим меню обираємо потрібне зображення. Після підтвердження кнопкою відкрити, його буде розміщено на поверхні початкового вікна. За допомогою кнопки Area Selection, можна увімкнути інструмент виділення певної частини зображення. Виділення виконується за допомогою нажаття і утримання лівої клавіші миші та пересування курсору зображенням. Цей процес супроводжується створенням і розширенням або звуженням рамки червоного кольору, це зображено на рисунку 3.3. Останнім кроком буде натиснути кнопку Segmentation. Зачекавши деякий час, ми отримаємо оброблене зображення

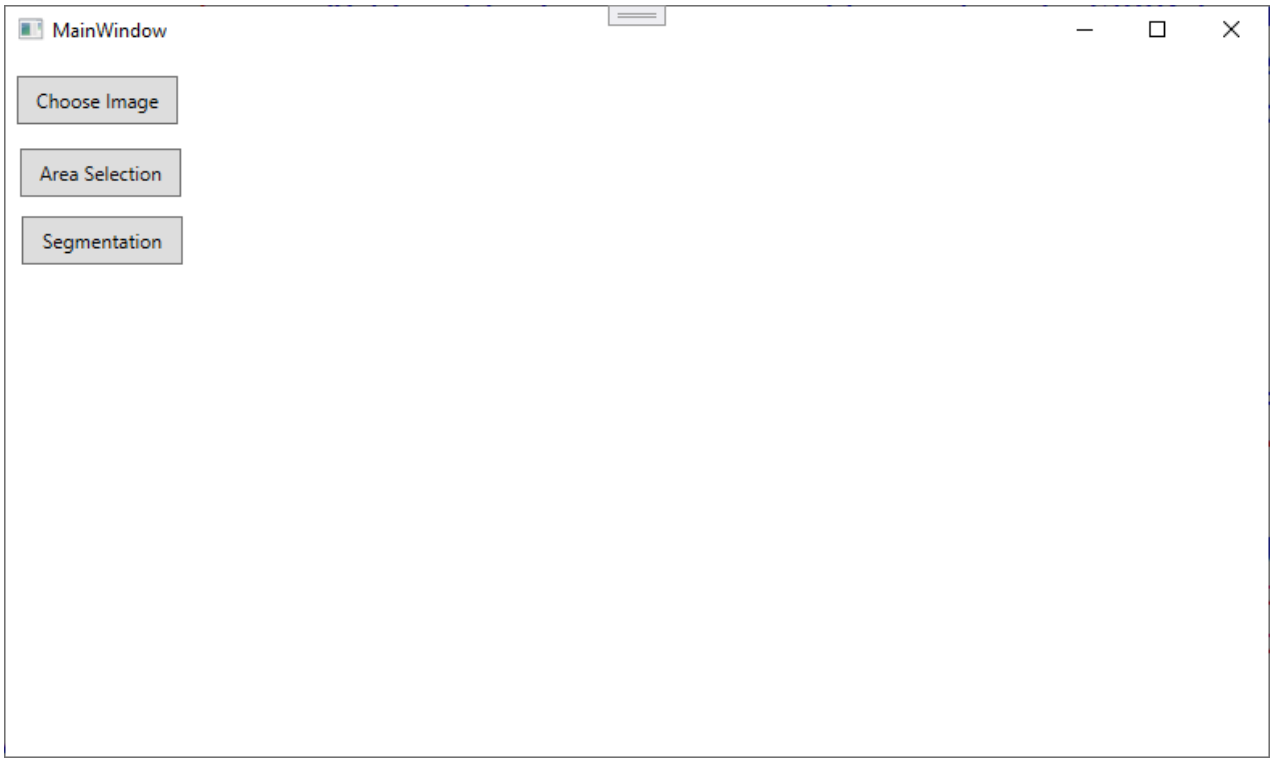

Рисунок 3.3 - Стартове вікно

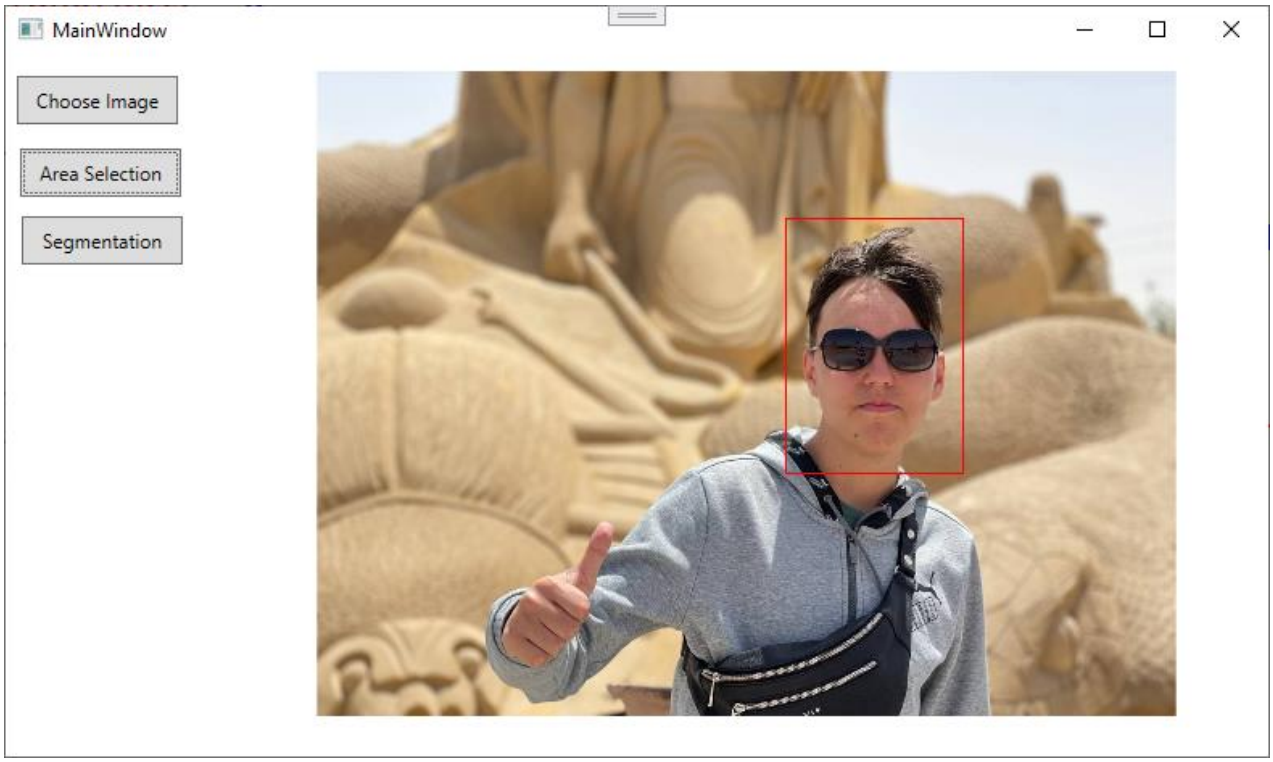

Рисунок 3.4 - приклад виділення об’єкта на зображенні

3.5 Тестування розробленої моделі

Маючи готовий застосунок, ми повинні провести тестування щоб впевнитись у коректності його роботи. Для цього нам знадобиться деяка вибірка зображень, у додатку А буде наведено їх початковий варіант. На виході ми повинні отримати вхідне зображення на якому знайдені та виділені знайомі для мережі об’єкти, на рисунку 3.4 видно що алгоритм зміг знайти: людину, годинник, ліжко, пляшку (і навіть її відображення у дзеркалі). На риснку 3.5 алгоритм теж знайшов людину.

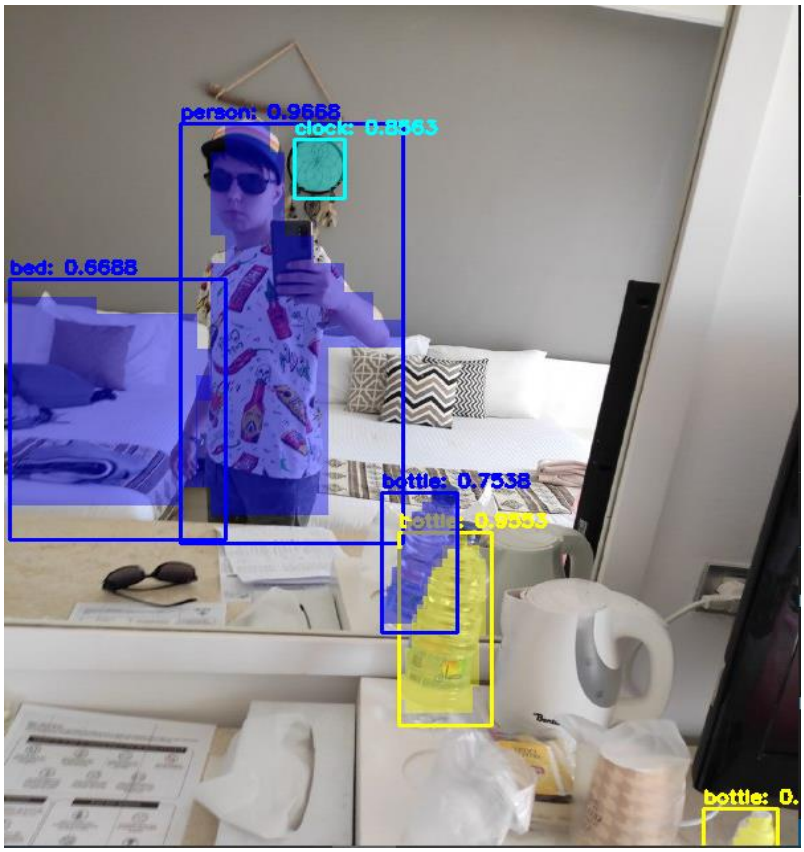

Рисунок 3.5 - Результат обробки зображення мережею Mask R-CNN

Рисунок 3.6 – Результат сегментації за допомогою фільтра grabcut

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований застосунок для сегментації зображень за допомогою згорточної нейронної мережі Mask R-CNN та Grabcut.

Отже сегментація зображень є важливою галуззю у всесвітньому розвитку комп'ютерних технологій. Вона надає змогу різноманітним застосункам розуміти вміст зображень і використовувати для своїх потреб. Останнім часом все більше областей потребує використання цієї технології для отримання правильних результатів та автоматизації деяких процесів. На сьогодні, різними авторами було розроблено безліч алгоритмів та підходів до цієї технології, що дозволяє нам дуже гнучко обрати той метод який відповідає нашим вимогам. Відповідно до наших потреб ми можемо використати як простий скрипт що розділить зображення на частини за певним правилом, так і нейронну мережу яка здатна коректно виділяти навіть складні об'єкти.

Результати роботи апробовано у вигляді статті під час III Міжнародної науково-практичної конференції «Scientific researches and methods of their carrying out: world experience and domestic realities» [10].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кобилін, О. А., & Творошенко, І. С. (2021) Методи цифрової обробки зображень: навч. посібник. *Харків: ХНУРЕ*.

2. Гороховатский В. А., (2014). Структурный анализ и интеллектуальная обработка данных в компьютерном зрении

3. Путятін, Є. П., Гороховатський, В. О., & Матат, О. О. (2006) Методи та алгоритми комп'ютерного зору: навч. посібник.

4. Кузьомін, О.Я., Василенко О.О. (2019) Моделювання у процесі проектування інтелектуальної медичної системи діагностування, *Радіоелектроніка та інформатика*, 2, pp. 61 – 66

5. Кузьомін О.Я., Василенко О.О., Свістунов І.О. (2020) Розробка багатоагентних структур для вирішення проблем медичної системи діагностування, *Радіоелектроніка та інформатика*, 2, pp. 47 – 54.

6. Кузьомін О.Я., Василенко О.О., Горшколєпов А.В. (2020) Розробка структур медичних агентів для вирішення проблем медичної системи діагностування, *Радіоелектроніка та інформатика*, 2, pp. 55 – 65.

7. Kuzomin O., Vasilenko O., Maliar B. (2019), Research of the intellectual system of knowledge search in Databases, *International Journal Information Models and Analyses*, Volume 7, Number 4., pp. 327 – 338.

8. Kuzomin O., Vasilenko O., Khripushina T. (2019), Intellectual Models and Means of the Biometric System Dynamics of Rinosinusite, *International Journal Information Models and Analyses*, Volume 7, Number 4. pp. 350 – 361.

9. Kuzomin O., Vasilenko O., Shapoval O. (2019), 2 Research of medical diagnostic data search methods, *International Journal Information Models and Analyses*, Volume 7, Number 4. pp. 339 – 349.

10. Terebetskyi M. A., Kuzomin O. Y. (2022) Development and research of image segmentation using mask r-cnn, grabcut and opencv, *International scientific journal «Grail of Science»*, 14/15, pp. 362-368.

11. Tushniskyi R. B., Kojuh I. Y. (2013) Investigation of parallel image segmentation algorithms using GPU calculations, *Eastern European Journal of Advanced Technology*, 2/4(62), pp. 59-64.

12. Volodin D., Afanasieva I. (2019) Analysis of methods for segmenting images of car registration numbers, *Bionics of intelligence*, 1(92), pp. 20-25.

13. Daradkeh, Y.I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L.A., and Ahmad, N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.

14. Daradkeh, Y.I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., and Al-Dhaifallah, M. (2021) Methods of Classification of Images on the Basis of the Values of Statistical Distributions for the Composition of Structural Description Components, *IEEE Access*, 9, pp. 92964-92973.

15. Gorokhovatskyi, V.O., Tvoroshenko, I.S., and Peredrii O.O. (2020) Image classification method modification based on model of logic processing of bit description weights vector, *Telecommunications and Radio Engineering*, 79(1), pp. 59-69.

16. Kuzomin O., Vasilenko O., Mertsalov A. (2019) Developing methods based on text mining technology to Improve the quality and speed of automatic clustering of Documents, I*nternational Journal Information Models and Analyses*, Volume 7, Number 4 pp. 385 – 397.

17. Kuzomin, O., Dudka, O., Vasylenko, O., Lyashenko, V. (2020) The patient organism modeling for diagnosis with the usage of a multi agent representation, *International Journal of Emerging Trends in Engineering Research*, 8(9), pp. 5733-5739

18. Kuzomin, O., Dudka, O., Vasylenko, O., Radchenko, V., Lyashenko, V 3Using of ontologies for building databases and knowledge bases for consequences management of emergency, *International Journal of Advanced Trends in Computer Science and Engineering*, 2020, 9(4), pp. 5040-5045

19. Canny J. A (1986.) Computational Approach to Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 8, Number 6, pp. 679–698

20. He K., Zhang X., Ren S., Sun J (2016) Deep Residual Learning for Image Recognition, *Proceedings of the CVPR*, pp. 70–78.

21. Gorokhovatskyi, V., Gorokhovatskyi, O., Yevgenyi, P., & Olena, P. (2018). Quantization of the Space of Structural Image Features as a Way to.

22. Tvoroshenko, I., & Kukharchuk, V. (2021). Current state of development of applications for recognition of faces in the image and frames of video captures.

23. Image Processing with Python: Image Segmentation using Thresholding Methods. URL: https://medium.com/swlh/image-processing-with-python-image-segmentation-using-thresholding-methods-423ecdaf8ab4 (дата звернення: 07.05.2022).

24. Image Processing And Pattern Recognition URL: http://ippr-practical.blogspot.com/2012/04/region-segmentation.html (дата звернення: 08.05.2022).

25. An Optimal Edge Detection using Gravitational Search Algorithm URL: https://www.researchgate.net/publication/282611684_An_Optimal_Edge_Detection_using_Gravitational_Search_Algorithm (дата звернення: 08.05.2022).

26. Image Segmentation using K Means Clustering URL: https://www.geeksforgeeks.org/image-segmentation-using-k-means-clustering/ (дата звернення: 09.05.2022).

27. Fully Convolutional Networks for Semantic Segmentation URL: https://www.cv-

foundation.org/openaccess/content_cvpr_2015/papers/Long_Fully_Convolutional_Networks_2015_CVPR_paper.pdf (дата звернення: 10.05.2022).

28. Object Detection. Recognize and rule. Part 2 URL: https://habr.com/ru/company/jetinfosystems/blog/498652/ (дата звернення: 17.05.2022).

29. Review: FCN — Fully Convolutional Network (Semantic Segmentation) URL: https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1 (дата звернення: 18.05.2022).

30. Faster R-CNN for object detection URL: https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46(дата звернення: 19.05.2022).

31. Mask R-CNN: Modern Neural Network Architecture for Object Segmentation in Images URL: https://habr.com/ru/post/421299/ (дата звернення: 19.05.2022).

32. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms URL: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e (дата звернення: 20.05.2022).

33. Рассел С., Норвіг П. (2009). Штучний інтелект: сучасний підхід: навч. посібник.

34. How U-net works? URL: https://developers.arcgis.com/python/guide/how-unet-works/ (дата звернення: 25.05.2022).