

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:
«Інформаційна технологія заміни рекламних банерів на відео»

Виконав: студент 2-го курсу, групи 1КН-21м
спеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Діденко Ю.В.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

Колесницький О.К.

(прізвище та ініціали)

« _____ » _____ 2022 р.

Опонент: к.т.н., професор каф. ПЗ

Паламарчук Є.А.

(прізвище та ініціали)

« _____ » _____ 2022 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)


« _____ » _____ 2022 р.

Вінниця ВНТУ – 2022 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.

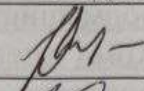
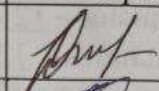


 14. 09. 2022 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТА

Діденко Юрій Володимирович
(прізвище, ім'я, по батькові)

1. Тема роботи: Інформаційна технологія заміни рекламних банерів на відео
керівник роботи: к.т.н., доцент кафедри КН Колесницький О.К.
затверджені наказом вищого навчального закладу від «14» 09 2022 року № 203
2. Строки подання студентом роботи 11. 11. 2022 року
3. Вихідні дані до роботи:
вхідні дані – мова об'єктно-орієнтованого програмування, браузерне середовище роботи додатку, кількість критеріїв класифікації = 7, вхідний відео-потік, обсяг навчальної вибірки не менше 2500, доступність додатку за посиланням, відповідність стандарту ODBS
4. Зміст текстової частини:
Вступ, аналіз предметної області, аналіз існуючих рішень, обґрунтування методу розв'язання задачі, обґрунтування вибору інструментів технічної реалізації, проектування інформаційної технології заміни рекламних банерів на відео, програмна реалізація інформаційної технології заміни рекламних банерів на відео, економічна частина, аналіз результатів тестування, розробка інструкції користувача.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)
Загальна структурна схема інформаційної технології, діаграма класів інформаційної технології, схема алгоритму роботи технології заміни рекламних банерів на відео, діаграма послідовності, графіки навчання нейронної мережі, загальний вигляд інтерфейсу користувача, приклад роботи нейронної мережі.

6. Консультанти розділів роботи


Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Колесницький О.К., к.т.н., доц. каф. КН		
5	Нікіфорова Л. О., к. е. н., доц. каф. ЕПВМ		

7. Дата видачі завдання _____ 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Очікувані результати
1	Аналіз сучасного рівня інформаційних технологій розпізнавання рекламних банерів. Постановка задачі дослідження	14.09.22	Аналітичний огляд літературних джерел, задачі досліджень, розділ 1
2	Побудова алгоритму дій заміни рекламних банерів на відео та реалізація компонентів програми інформаційної технології.	02.10.22	Моделі, розділ 2, 3
3	Практичне застосування та оцінка ефективності розробленої програми	17.10.22	розділ 3, 4
4	Підготовка економічної частини	08.11.22	розділ 5
5	Апробація та/або впровадження результатів дослідження	23.11.22	тези доповідей
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	07.11.22	Пояснювальна записка, графічний матеріал, презентація

Студент


(підпис)

Діденко Ю.В.

Керівник роботи

(підпис)

Колесницький О.К.

АНОТАЦІЯ

УДК 004.8

Діденко Ю.В. Інформаційна технологія заміни рекламних банерів на відео. Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма – комп'ютерні науки. Вінниця: ВНТУ, 2022. 98с.

На укр. Мові. Бібліогр.: 30 назв; рис.: 24; табл. 10

Дана магістерська кваліфікаційна робота присвячена розробці програмного забезпечення для автоматизованої заміни рекламних банерів у відео. В даній роботі було досліджено відомі методи сегментації й вбудовування зображень та проаналізовано способи їх реалізації. Було обрано нейро-мережевий метод та відповідні механізми навчання нейронних мереж. Для вбудовування зображення в сегментовану маску, використовуються класичні алгоритми комп'ютерного зору. Архітектура обраної нейронної мережі базується на архітектурі мережі Mask-RCNN, з доповненими шарами класифікації та згортки. Вхідними даними до мережі є зображення розміром 512 на 512 пікселів, вихідним значенням є вектор розмірністю 1 на 10, що позначає категорії рекламних банерів. Для програмної реалізації було обрано мову програмування Python та бібліотеку TensorFlow. Достовірність розпізнавання рекламних банерів на 10% краще за аналоги, а точність встановлення рекламних банерів на 18%. Загальне покращення достовірності заміни рекламних банерів на 4%.

Графічна частина складається з 7 плакатів.

У економічному розділі розраховано суму витрат на розробку та виготовлення нового технологічного рішення, яка склала 559628 гривень, прогнозована орієнтована величина витрат по кожній з статей витрат, розраховано чистий прибуток, термін окупності витрат для виробника 0.92 роки та економічний ефект для споживача при використанні даної розробки.

Ключові слова: розпізнавання, сегментація зображень, нейронна мережа, згорткові мережі, комп'ютерний зір.

ABSTRACT

Didenko Yu.V. Information technology for replacing advertising banners with video. Master's qualification thesis on specialty 122 - computer science, educational program - computer science. Vinnytsia: VNTU, 2022. 98p.

In Ukrainian Languages Bibliography: 30 titles; Fig.: 24; table 10

This master's thesis is devoted to the development of software for automated replacement of advertising banners in videos. In this work, known methods of image segmentation and embedding were investigated and methods of their implementation were analyzed. The neural network method and corresponding neural network learning mechanisms were chosen. Classic computer vision algorithms are used to embed the image into the segmented mask. The architecture of the selected neural network is based on the architecture of the Mask-RCNN network, with added layers of classification and convolution. The input data to the network is a 512 by 512 pixel image, the output value is a 1 by 10 vector representing the categories of advertising banners. Python programming language and TensorFlow library were chosen for software implementation. The reliability of recognizing advertising banners is 10% better than analogues, and the accuracy of installing advertising banners is 18%.

The graphic part consists of 7 posters.

In the economic section, the amount of costs for the development and production of a new technological solution was calculated, which amounted to 559,628 hryvnias, the projected estimated amount of costs for each of the cost items, the net profit, the payback period for the manufacturer of 0.92 years and the economic effect for the consumer when using this development were calculated.

Keywords: recognition, image segmentation, neural network, convolutional networks, computer vision.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАМІНИ РЕКЛАМНИХ БАНЕРІВ НА ВІДЕО	10
1.1 Постановка задачі заміни рекламних банерів на відео	10
1.2 Огляд відомих методів заміни рекламних банерів на відео та обґрунтування вибору нейромережевого методу	12
1.3 Обґрунтування вибору аналогу до інформаційної технології заміни реklamних банерів на відео	17
1.4 Висновок до розділу 1	20
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЗАМІНИ РЕКЛАМНИХ БАНЕРІВ НА ВІДЕО	21
2.1 Обґрунтування вибору нейронної мережі	21
2.2 Структура та функціонування згорткової нейронної мережі	27
2.3 Особливості процесу виявлення рекламних банерів на відео	30
2.4 Формування навчального набору	31
2.5 Розробка алгоритму заміни рекламних банерів на відео	32
2.6 Розробка UML-діаграми класів	35
2.7 Висновок до розділу 2	38
3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЗАМІНИ РЕКЛАМНИХ БАНЕРІВ НА ВІДЕО	39
3.1 Обґрунтування вибору мови та середовища програмування для реалізації програми заміни рекламних банерів на відео	39
3.2 Вибір програмних інструментів	41
3.3 Навчання нейронної мережі	43
3.4 Програмна реалізація інформаційної технології заміни рекламних банерів на відео	45
3.5 Висновок до розділу 3	53
4 ТЕСТУВАННЯ ТА АНАЛІЗ РОБОТИ ПРОГРАМИ ЗАМІНИ РЕКЛАМНИХ БАНЕРІВ НА ВІДЕО	54
4.1 Тестування роботи інформаційної технології заміни рекламних банерів на відео	54
4.2 Порівняння з аналогами	57
4.3 Висновок до розділу 4	59
5 ЕКОНОМІЧНА ЧАСТИНА	60

	5
5.1 Комерційний та технологічний аудит науково-технічної розробки....	60
5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи.....	63
5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	69
5.4 Висновок до розділу 5	74
ВИСНОВКИ.....	75
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	77
ДОДАТКИ.....	80
Додаток А (обов'язковий) Результат перевірки на плагіат в онлайн-системі UNICHECK	81
Додаток Б (обов'язковий) Лістинг програми	82
Додаток В (обов'язковий) Ілюстративна частина	87
Додаток Г (довідниковий) Інструкція користувача.....	95

ВСТУП

Актуальність. В житті сучасної людини поглинання медіа-контенту є невід'ємною частиною проведення вільного часу. До того ж, споживачі контенту сприяють швидкому розвитку даного напрямку. Медіа-контент в свою чергу поділяється на фото, відео та аудіо інформацію. Оскільки, відео-контент в даний період набув найбільшої популярності, через поєднання візуальної та аудіо інформації, було обрано саме взаємодію з відео інформацією. Зокрема, відео-контент використовується як в інформаційній так і в розважальній сфері.

Неабияку роль відео-контент відіграє в рекламних кампаніях. Медіа завжди відіграло провідну роль в просуванні товарів, а з ростом попиту на якісну цільову рекламу, змінилися й підходи до формування маркетингових кампаній [1]. Щодня генерують тисячі нових відео, більша частина з яких містить рекламну інтеграцію. Звичайно, що окремі категорії відео мають вдало продумано рекламну кампанію та отримують максимальний прибуток від інтеграції, проте існують галузі які не можуть скористатись маркетинговими можливостями на повну. Наприклад, футбольні матчі, де є можливість представлення реклами на обмеженій площі, навколо поля. В таких випадках, рекламні агенції намагаються розмістити рекламу, що є найбільш актуальною для великої кількості людей.

Уявіть, що відео-контент генерується в Каліфорнії, США, і це мають засвідчити глядачі в Токіо, Японія. Демографічні показники та звички споживачів у Каліфорнії та Токіо значно відрізняються. Щоб забезпечити вражаючий перегляд глядачів у Токіо, рекламу в Інтернет-відео потрібно ретельно підготувати.

Основний недолік використання такого підходу, це прив'язування контексту відео, до певних брендів та компаній, що можуть собі дозволити дороговартісну промоцію [2]. Сприйняття цієї реклами може відрізнятись в залежності від середовища дистрибуції відео-контенту. Звідси виникає

потреба у створенні інструментів, що дозволяють підлаштовувати рекламні кампанії, під конкретного користувача, на локальних рівнях дистрибуції медіа продукту.

Зв'язок роботи з науковими програмами, планами, темами.

Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання досліджень. Мета магістерської кваліфікаційної роботи - підвищення достовірності виявлення та заміни рекламних банерів на відео.

Для досягнення мети роботи потрібно виконати такі задачі:

- здійснити аналіз предметної області сегментації рекламних банерів на відео;
- розглянути існуючі методи сегментації рекламних банерів на відео та обґрунтувати обраний метод, що задовольняє мету даної магістерської роботи;
- розробити модель використання нейронних мереж для сегментації рекламних банерів на відео;
- сформулювати стадії інформаційної технології, розробити алгоритм роботи програмного забезпечення;
- здійснити програмну реалізацію запропонованої інформаційної технології;
- провести тестування програмного засобу та проаналізувати отримані результати.
- обрахувати економічну доцільність розробленого програмного засобу

Об'єкт дослідження – процес заміни рекламних банерів на відео.

Предмет дослідження – програмні засоби, що виконують комп'ютеризований процес заміни рекламних банерів на відео.

Методи дослідження. У роботі використані такі методи наукових досліджень: методи системного аналізу, теорія штучних нейронних мереж для реалізації інформаційної технології, методи математичної статистики для розробки процесу розпізнавання математичних формул, методи оцінки достовірності їх роботи та обрахунків результатів експериментів із програмним забезпеченням, об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів.

Набула подальшого розвитку інформаційна технологія заміни рекламних банерів на відео, яка відрізняється використанням сегментаційної глибинної нейронної мережі прямого поширення, що дозволило підвищити достовірність роботи програмних засобів виявлення та заміни рекламних банерів на відео.

Практичне значення одержаних результатів полягає в тому, що на основі проведених досліджень було розроблено програмне забезпечення для виявлення та заміни рекламних банерів на відео.

Запропонована інформаційна технологія сприяє підвищенню швидкодії і достовірності виявлення та заміни рекламних банерів на відео, зокрема:

- розроблено алгоритм роботи програмного забезпечення заміни рекламних банерів на відео;
- розроблено програмні засоби для виявлення рекламних об'єктів на відео на основі штучної нейронної мережі;
- розроблено програмні засоби для вбудовування зображення в сегментовану маску реклами.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційних технології заміни рекламних банерів на відео. Адекватність

розроблених математичних моделей підтверджується результатами експериментальних досліджень.

Особистий внесок здобувача. Усі результати наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, які написано у співавторстві, здобувачу належить: аналіз процесу розв’язання задачі виявлення рекламних банерів та вбудовування реклами у маску сегментації.

Апробація результатів роботи. Результати досліджень апробовані на конференції «Молодь в науці: дослідження, проблеми, перспективи-2023», Вінниця, листопад 2022 – травень 2023 року [1].

Публікації. За результатами досліджень опубліковано одні тези доповіді на науково-технічній конференції.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАМІНИ РЕКЛАМНИХ БАНЕРІВ НА ВІДЕО

1.1 Постановка задачі заміни рекламних банерів на відео

Віртуальна відео реклама є важливою і перспективною областю в галузі інтернет-реклами. Вона включає інтеграцію реклами у прямі трансляції або записані відео для розміщення продуктів та цільової реклами. Така інтеграція в основному здійснюється відео редакторами на етапі пост продукції, що є громіздким і трудомістким процесом [3, 4]. Простір для кандидатів повинен відповідати перспективі сцени, а також вимагає високої якості імплементації враховуючи суб'єктивні судження людини. Тому, важливо автоматично визначати потенційні місця для кандидатів у відеокадрі, куди можна імпантувати нові рекламні оголошення.

Можливість автоматичного виявлення потенційних просторів, де можна розмістити рекламу, не заважаючи глядачеві або потоку сцени є надзвичайно важливою [3].

Уявіть, що ви дивитесь своє улюблене телевізійне шоу і під час конкретної сцени, ви бачите рекламу улюбленого автомобіля чи смартфона біля дороги. Подібно до цього, ваш друг, який проживає в іншій частині земної кулі, дивлячись одне і те ж телевізійне шоу, бачить іншу рекламу, своєї улюбленої музичної групи, яка має популярність у місці його проживання. Такі цільові рекламні шляхи відкривають потенціал генерації персоналізованого відео контенту для різних демографічних груп [4]. Нова реклама може бути інтегрована на різних об'єктах, наприклад, на стіні, вже існуючому рекламному банері, однотонному одязі, тощо.

Існує, потенційно, великий ринок, що зацікавлений в пришвидшенні створенні реклами та відокремленню рекламних об'єктів від конкретних рекламних кампаній та від людини [5, 6]. Так, рекламні банери можна вбудовувати у такі сцени відео:

- вже вбудовані рекламні інтеграції
- площини, де є можливість чітко окреслити багатогранні геометричні фігури
- однотонний одяг
- предмети інтер'єру з чітким контуром
- задній фон, у вигляді водяного знаку

Вищенаведений список є далеко не повним, але відображає широкий спектр можливостей для подальших досліджень даної сфери. Серед наведених пропозицій, найбільш оптимальним є варіант імплементації реклами у вже вбудовану. Оскільки, цей процес вже був заздалегідь розрахованим маркетологами, тобто певні місця у кадрі найбільше привертають увагу глядачів, отже ефект від такої інтеграції є найкращим.

Проте, це вимагає значних ресурсів, спеціально навчених дизайнерів, що значно підвищує вартість інтеграції такої реклами у кожне відео та не є економічно вигідним для замовників. Зокрема, як вже згадувалось вище, показник якості ручної заміни відрізняється в залежності від суб'єктивних суджень кожного глядача, отже немає стандартизованого та універсального рішення, яке вирішує дану проблему. В подальших дослідженнях можна проаналізувати залежність успіху рекламної кампанії від різних параметрів рекламних банерів[4, 6].

Здешевлення процесу імплементації рекламних банерів у медіа контент підіграло б інтерес компаній до технологій машинного навчання у сфері маркетингу та сприяло б розвитку цієї сфери. Тому було поставлено задачу, розробити програму заміни рекламних банерів на відео, що дозволяє автоматизувати процес виявлення рекламних банерів та їх заміни у відеокадрі.

1.2 Огляд відомих методів заміни рекламних банерів на відео та обґрунтування вибору нейромережевого методу

Проблему заміни реклами досліджують безліч студентів-науковців у всьому світі, оскільки сфера доволі перспективна та має зацікавлених сторін [5, 7]. Вирішення поставленої задачі вимагає комплексного підходу, оскільки перш ніж імплементувати рекламу, потрібно точно визначити область, де буде відбуватись заміна. Власне, задача пошуку місця для вбудовування реклами і є ключовим елементом програми.

Для нативності вбудованої реклами необхідно використати не просто виявлення рекламних банерів, а сегментувати область де знаходиться банер. З розвитком області комп'ютерного зору, було виділено 3 найефективніших методи обробки зображень, що дозволяють сегментувати об'єкти у кадрі. Перший та базовий – це опрацювання зображення за допомогою математичних формул та функцій, ці підходи ще називають класичними алгоритмами комп'ютерного зору. Інший підхід, що застосовується над математичними функціями, але є відокремленою областю досліджень – машинне навчання. Алгоритми машинного навчання використовують оптимізаційно-ітеративні, а не аналітичні підходи, щодо вирішення задач комп'ютерного зору, зокрема сегментації чи виявлення об'єктів. Еталонним та найсучаснішим вважається третій метод обробки зображень, а саме штучні нейронні мережі. Даний метод в певній мірі є розширенням двох попередніх, але є дещо складнішим, тому способи вирішення задач відрізняються форматом представлення інформації, що опрацьовується [5, 8, 9].

Сегментація зображень спочатку почалася з цифрової обробки зображень у поєднанні з алгоритмами оптимізації. Ці примітивні алгоритми використовували такі методи, як вирощування області та алгоритм змій, де вони встановлювали початкові області, а алгоритм порівнював значення пікселів, щоб отримати уявлення про карту сегментів. Ці методи

використовували локальний погляд на особливості зображення та зосереджувалися на локальних відмінностях і градієнтах у пікселях.

Алгоритми, які мали глобальний погляд на вхідне зображення, з'явилися набагато пізніше, коли серед класичних методів обробки зображень були запропоновані такі методи, як адаптивне порогове визначення, алгоритм Оцу та алгоритми кластеризації.

Кожен з запропонованих методів має свої переваги та недоліки, які потрібно чітко визначити, щоб обрати найбільш оптимальний метод для вирішення задачі. Серед запропонованих методів можна виокремити такі алгоритми сегментації зображень:

1.2.1 Сегментація країв

З точки зору сегментації ми можемо сказати, що виявлення країв відповідає класифікації пікселів у зображенні як краєві пікселі та виділенню цих крайових пікселів відповідно до окремого класу.

Виявлення країв зазвичай виконується за допомогою спеціальних фільтрів, які дають нам краї зображення після згортання. Ці фільтри обчислюються спеціальними алгоритмами, які працюють над оцінкою градієнтів зображення в координатах x і y просторової площини.

Щоб виконати сегментацію країв, необхідно, застосувати оператор Собеля та Лапласа, тобто застосувати фільтри до зображення. Фільтр Собеля поділяється на вертикальний, значення якого зменшуються зліва на право та горизонтальний, який поширюється зверху донизу. Оператор Собеля являє собою фільтр розмірністю $3 \times 3 \times 1$, що містить в одній частині позитивні значення, а в іншій стороні негативні значення, по середині фільтру знаходяться нулі. Для прикладу на рисунку 1.1 зображено вертикальний оператор Собеля. Оператор Лапласа це фільтр розмірністю $N \times N \times 1$, що в центрі містить пікове значення, а збільшуючи відстань від центру, значення в комірці відповідно зменшуються.

$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

Рисунок 1.1 – Вертикальний оператор Собеля

1.2.2 Сегментація на основі кластеризації

Сучасні процедури сегментації на основі машинного навчання, які залежать від методів обробки зображень, зазвичай використовують для сегментації алгоритми кластеризації.

Алгоритми кластеризації працюють краще, ніж аналоги, і можуть забезпечити достатньо якісні сегменти за невеликий проміжок часу. Популярні алгоритми, такі як алгоритми кластеризації К-середніх, є неконтрольованими алгоритмами, які працюють шляхом кластеризації пікселів із загальними атрибутами разом як належних до певного сегмента.

К-означає кластеризацію, зокрема, бере до уваги всі пікселі та об'єднує їх у класи «k». На відміну від методів нарощування регіонів, методам на основі кластеризації не потрібна початкова точка для початку сегментації.

Оскільки, в контексті сегментації зображень, кластеризація враховує найпримітивніші характеристики такі як колір об'єкту чи відтінок, якість таких моделей сегментації залишається доволі низькою.

1.2.3 Методи на основі глибокого навчання

Моделі семантичної сегментації надають карти сегментів як виходи, що відповідають входам, які вони подають.

Ці карти сегментів часто є n-канальними, де n означає кількість класів, які модель повинна сегментувати. Кожен із цих n-каналів є двійковим за своєю природою, де розташування об'єктів «заповнено» одиницями, а порожні області складаються з нулів. Основна карта істинності — це одноканальний масив цілих чисел такого ж розміру, як і вхід, і має діапазон «n», причому

кожен сегмент «заповнений» значенням індексу відповідних класів (класи індексуються від 0 до $n-1$). Вихід моделі в “ n -канальному” двійковому форматі також відомий як двовимірне представлення прогнозів у однократному кодуванні, приклад такого виходу зображено на рисунку 1.2. Нейронні мережі, які виконують сегментацію, зазвичай використовують структуру кодер-декодер, де кодер супроводжується вузьким місцем і декодером або рівнями підвищення дискретизації безпосередньо з вузького місця (як у FCN).

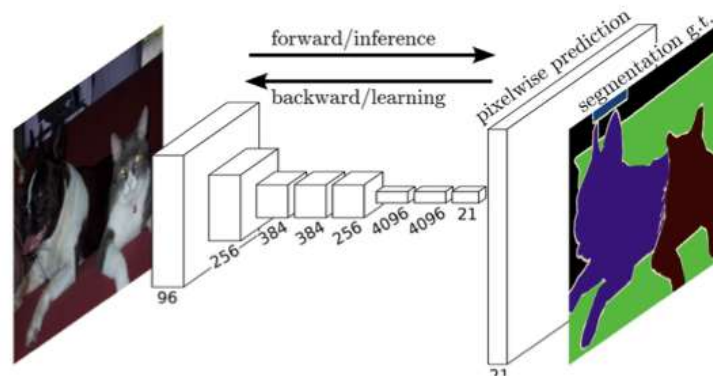


Рисунок 1.2 – Приклад маски виходу сегментаційної нейронної мережі

Визначені методи виконують певні обчислення над зображенням та в результаті опрацювання вхідних даних, повертають вихідне значення, що представляє набір точок. Залежно від методу, вихідне значення може мати різний формат, наприклад метод математичних моделей, повертає набір координат, що являють собою кути геометричної фігури, таким чином формують область, яку обмежують прямими лініями зі всіх сторін. В свою чергу нейромережевий метод може повертати як набір координат, так і набір точок, де кожна точка відповідає області, куди буде відбуватись імплементація промоції [5, 7, 9].

Математичними моделями розпізнавання об'єктів виконується шляхом попереднього зіставлення кожної ключової точки зображення з заздалегідь заданим паттерном, базуючись на аналітичних рівняннях. Багато з цих початкових збігів будуть неправильними через неоднозначні особливості або

особливості, що виникають внаслідок безладу. Тому спочатку ідентифікуються кластери щонайменше з 3 об'єктів, які узгоджують об'єкт та його позицію, оскільки ці кластери мають набагато більшу ймовірність бути правильними, ніж окремі збіги об'єктів. Потім кожен кластер перевіряється, виконуючи детальну геометричну підгонку до моделі і результат використовується для прийняття або відхилення інтерпретації. В результаті, точність таких методів, доволі слабка, тому що формули мають певні математичні обмеження, та не здатні знаходити складні залежності й не є стійкими до зміни розміру зображення [8, 10].

Алгоритми машинного навчання мають схожий підхід до вирішення задачі, але дещо відрізняються необхідністю попереднього навчання. Для виявлення рекламних банерів можна використовувати бінарні класифікатори в композиції з деревами рішень, що дозволяє визначити декілька класів. Використання даного підходу вимагає доволі великої навчальної вибірки, щоб збільшити здатність моделі до узагальнення. Для того щоб застосувати певний класифікатор, потрібно отримати характеристики зображення, в конкретній задачі характеристикою виступає гістограма направлених градієнтів зображення. Для навчання алгоритму використовується частина зображення як паттерн, який позначається двома класами, такий що належить зображенню та не належить. Метод рухомого вікна дозволяє циклічно пройтись частинами зображення, в місці де кількість збігів перевищує певний коефіцієнт, присвоюється мітка класу виявленого об'єкта [9, 11]. Даний підхід, є значно кращим ніж класичні математичні обчислення, але вимагає значних зусиль під час навчання та розмітки зображення. І також не гарантує високої точності, тому що це все таки розширені математичні моделі, що мають ті ж обмеження.

З ростом обчислювальних можливостей почали стрімко розвиватись нейронні мережі. Щороку проводяться десятки наукових конференцій та конкурсів присвячених нейронним мережам, в області комп'ютерного зору. В даний момент часу, нейронні мережі вважаються найсучаснішим та найкращим методом опрацювання зображень. Вони здатні вирішувати

визначену проблему за допомогою двох підходів, таких як сегментація та виявлення об'єкта у кадрі. Виявлення об'єкту базується на визначенні координат прямокутника, який охоплює потрібний об'єкт, з заданою точністю [8]. В свою чергу, сегментація є розширеною версією виявлення, оскільки повертає, як вихід – маску зображення, тобто масив, де кожна точка координатної сітки має мітку, що визначає приналежність даної точки класу. Точність сучасних нейронних мереж сягає 90-95%, що викликає інтерес в науковців й бізнесу та дозволяє вирішувати задачі різного призначення. Однак, така точність вимагає ретельного збору та підготовки даних, це як наслідок обумовлює довгий процес розробки таких мереж [9, 11, 12, 15].

Оскільки поставлена задача на магістерську кваліфікаційну роботу ставить в пріоритеті, максимальну точність заміни рекламних банерів, то саме нейромережевий метод, може забезпечити цю характеристику. У цій роботі пропонується використовувати нейронну мережу, яка може автоматизувати процес заміни реклами у відеокадрі.

1.3 Обґрунтування вибору аналогу до інформаційної технології заміни рекламних банерів на відео

Для вирішення поставленої задачі, необхідно проаналізувати існуючі рішення та обрати одне з них для порівняльної характеристики. Наразі на ринку цільової відеореклами існує лише декілька комерційних рішень, інші ж поки на етапі досліджень.

Одним з запропонованих рішень є сервіс британської компанії «toast». Рекламна агенція пропонує рішення по версіонуванні реклами, тобто створення декількох версій відеоролику з різними рекламними банерами. Проте, «toast» пропонує лише заміну етикеток, що мають великий розмір у кадрі, це в свою чергу, дещо суперечить контексту реклами, у випадках коли змінюючи перспективу камери та масштаб етикетки, вона буде різко відтворювати оригінальну картинку. Компанія використовує підхід 3d

трекінгу, що дозволяє відслідковувати розміщення етикетки в часі [13]. На рисунку 1.1 можна побачити результат роботи аналогу «toast». Зокрема, ще одним обмеженням для клієнтів сервісу, є те що він не доступний всім, компанія виконує такі заміну лише за індивідуальним замовленням. Таким чином обмежуючи доступ до технології для великої аудиторії. На жаль, у відкритих джерелах відсутня інформація, щодо технологій та інструментів, які використовуються сервісом, відповідно відсутні будь-які метрики, які б в подальшому дали можливість порівняти розроблену технологію з аналогами.



Рисунок 1.3 – Приклад роботи аналогу «toast»

Серед рішень ірландського дослідницького центру «Nuawei» є система, де потенційними кандидатами для заміни виступають біл-борди, що зустрічаються на відео фреймах [14, 15]. В своїй роботі, вони представляють рішення на основі нейронної мережі, що дозволяє вставляти потрібні промоційні банери у біл-борди. Технологія працює за принципом пошуку потенційних місць для вбудовування реклами у всі біл-борди, що придатні для цього. Основною метою дослідження, було проведення аналізу найбільш популярних мереж, які здатні вирішувати поставлену задачу найкраще. Науковці обрали 5 типів найбільш успішних архітектур нейромереж та порівняли їх між собою, при чому, всі з них були згортковими нейромережами, так як архітектура даного типу є еталонною для опрацювання медіа формату. Нейронна мережа локалізує розташування білборда, але не вставляє туди власну промоцію, тобто не вирішує кінцевої проблеми. Головною перевагою,

є те що в результаті було створено власну архітектуру мережі DeepAds, яка запозичила найкращі властивості від інших використаних у дослідженні мережах [15]. Однак, це рішення не дає готового результату, а лише виконує один з проміжних етапів задачі, досліджуваної в роботі.

Іншим прикладом є також дослідження, що спирається на результати попереднього, а саме використовує запропоновану найкращу мережу, але з іншими вхідними та вихідними даними. Науковці використали архітектуру DeepAds в контексті пошуку оптимального місця на площинах, для вбудовування банеру. На відміну від результатів першого дослідження, в цьому випадку мережа не локалізує наявність білбордів, а шукає найкращі місця для їх потенційного розташування, враховуючи контекст зображення [15].

Ще одним схожим за функціоналом рішенням, є сервіс «Amazon Rekognition». Він дозволяє аналізувати відео за багатьма параметрами, але існують певні обмеження, щодо використання сервісу. Використання «Amazon Rekognition» вимагає постійного підключення до високошвидкісного інтернету та додаткових витрат для зберігання відео на серверах «Amazon» [16, 17]. Великим плюсом даного сервісу є велика кількість готових рішень, які можна комбінувати для вирішення потрібних задач, зокрема досліджуваних в роботі, проте для локальних задач, інфраструктура «Amazon» буде доволі громіздка та дороговартісна.

Можна побачити, що існуючі рішення можуть вирішувати поставлену задачу лише частково, або ж користувачам потрібно буде переплачувати значні суми за власне створення та використання подібних рішень. Тому створення компромісної технології може вирішити вищенаведені проблеми існуючих рішень.

1.4 Висновок до розділу 1

У першому розділі було проведено аналіз предметної області розв'язання задачі заміни рекламних банерів у відео, а саме – сформульовано постановку задачі, проведено огляд відомих методів вирішення задачі заміни рекламних банерів у відео. Також визначено актуальність проблеми заміни рекламних банерів у відео та описано основні проблеми існуючих рішень. Зокрема, проблеми даної сфери у відсутності стандартних підходів та універсальних рішень, що змогли б виконувати задачу заміни рекламних банерів у відео. На основі цього, можна сформулювати основну задачу, яка буде вирішуватись у даній роботі – розробка компромісного рішення, що дасть змогу виконувати заміну рекламних банерів у відео, й не вимагає втручання людини в цей процес. Для реалізації інформаційної технології було обрано найперспективніший метод на основі нейронних мереж. Автоматизована технологія, повинна самостійно визначати рекламні банери та вставляти туди необхідну користувачу інформаційної технології рекламу.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІНОЇ ТЕХНОЛОГІЇ ЗАМІНИ РЕКЛАМНИХ БАНЕРІВ НА ВІДЕО

2.1 Обґрунтування вибору нейронної мережі

Основною задачею в автоматизованому процесі заміни рекламних банерів є їх виявлення, у даному випадку – сегментація необхідного банеру.

Сегментація зображення – відноситься до процесу розподілу цифрового зображення на декілька сегментів, тобто набору пікселів, подібних за деякими критеріями однорідності, таких як колір, інтенсивність або текстура, з тим щоб знайти та ідентифікувати об'єкти та межі на зображенні [12, 17]. Точніше, сегментація зображень – це процес присвоєння таких міток кожному пікселю зображення, що пікселі з однаковими мітками мають спільні візуальні характеристики. Практичне застосування сегментації зображень – від фільтрації шумних зображень, медичного застосування, визначення об'єктів на супутникових знімках, розпізнавання обличчя, розпізнавання відбитків пальців, тощо. У літературі пропонується багато методів сегментації. Вибір методу сегментації над іншим та рівень сегментації визначаються типом зображення та характеристиками проблеми, що розглядається [18].

Сегментація зображень поділяється на 2 типи:

- семантична сегментація
- сегментація екземпляру

На рисунку 2.1 можна побачити результат опрацювання зображення сегментаційними нейронними мережами, що виконують об'єктну та семантичну сегментацію.

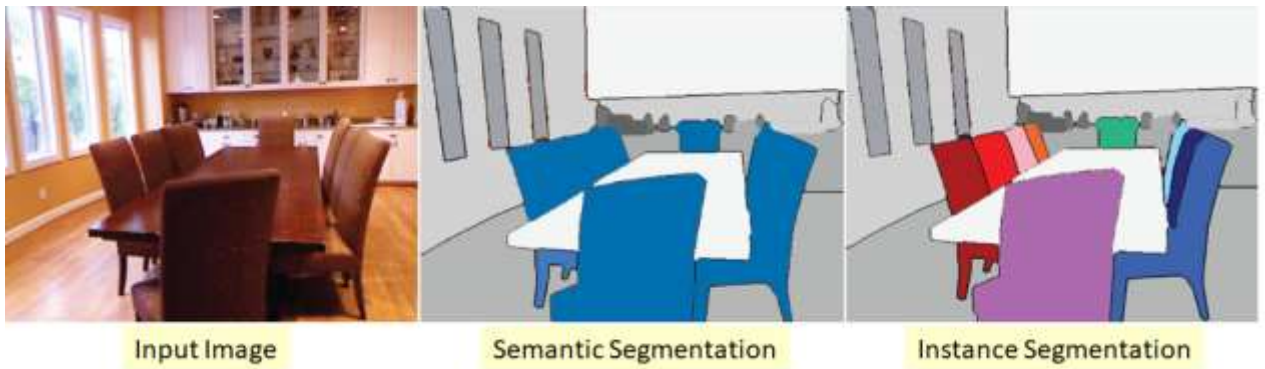


Рисунок 2.1 – Приклад вихідного зображення сегментаційних нейронних мереж

Для виконання сегментації за допомогою нейронних мереж використовуються так звані згорткові нейронні мережі, що моделюють функцію згортки в своїй архітектурі. Згорткові нейронні мережі під час навчання, оптимізують визначену кількість згорток шляхом мінімізації функції втрат. В свою чергу функція втрат визначається на основі поставленої задачі. Тому, навчання нейронної мережі даного типу зводиться до підбору правильних згорткових функцій, які в композиції можуть представляти контекст зображення в форматі цифр, які далі інтерпретуються програмним забезпеченням для виконання наступних обчислень [17]. Можна стверджувати, що цей тип нейронних мереж дозволяє комп'ютеру розуміти зміст зображення. Загальний вигляд архітектури сегментаційної нейронної мережі можна побачити на рисунку 2.2.

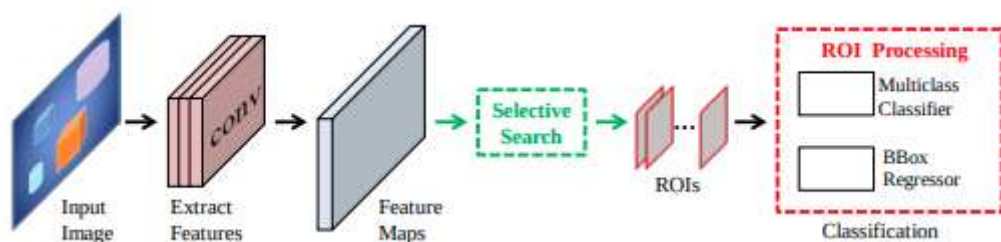


Рисунок 2.2 – Архітектура базової сегментаційної нейронної мережі

Семантична сегментація описує процес асоціювання кожного пікселя зображення з міткою класу. Ідея семантичної сегментації полягає у розробці техніки-алгоритму, яка добре працює в двох сферах, що забезпечує кращу точність сегментації та кращу ефективність сегментації. Краща точність сегментації охоплює точну локалізацію та розпізнавання класів на зображеннях, в результаті чого можна виділити велику різноманітність категорій, пов'язаних з об'єктами в реальному сценарії, а також випадки об'єктів, що належать до одного класу, які є предметом до внутрішньо класових змін зовнішнього вигляду, можуть бути локалізовані та розпізнані.

На відміну від семантичної сегментації, сегментація екземплярів незалежно маскує кожен екземпляр класу, що містить зображення. Завдання виявлення об'єктів та сегментації екземплярів досить корелюють. При виявленні об'єктів дослідники використовують обмежувальну рамку для виявлення кожного екземпляра об'єкта зображення з міткою для класифікації. Сегментація екземпляра ставить це завдання на крок вперед і встановлює маску сегментації для кожного екземпляра [9, 11].

У нашому випадку, сегментація екземпляра значно розширює можливості модуля, тому далі будемо розглядати архітектури нейромереж, що покривають задачу сегментації екземпляра. З появою глибокого навчання, більш конкретно, згорткових нейронних мереж (CNN), було запропоновано багато систем сегментації екземплярів, наприклад, в яких точність сегментації швидко зростала. Найбільший успіх отримала архітектура Mask R-CNN, саме її ми будемо розглядати, а також її покращений аналог Mask Scoring R-CNN [18, 19].

Mask R-CNN – це прямий та ефективний підхід до сегментації екземплярів. Беручи керівництво від Fast та Faster R-CNN, повністю з'єднана мережа (FCN) була використана для прогнозування масок сегментації, поряд з регресією та класифікацією об'єктів [18]. Для високої продуктивності функціональна пірамідальна мережа (FPN) була використана для вилучення поетапних мережевих функцій, в яких мережевий шлях зверху вниз, що має

бічні з'єднання, використовувався для отримання функцій, які є семантично надійними. На рисунку 2.3 можна побачити загальну структуру мережі. Деякі відносно нові набори даних забезпечують достатній простір для вдосконалення запропонованих методів. Загальні об'єкти в контексті Microsoft або набір даних COCO містять 200 тисяч зображень [18]. На зображеннях цього набору даних було зафіксовано багато випадків, що мають складний просторовий макет. Крім того, набір даних Cityscapes та набір даних Mapillary Vistas або MVD мають зображення вуличних сцен, що містять велику кількість об'єктів дорожнього руху на одне зображення. На зображеннях цих наборів даних виявляються випадки розмитості, оклюзії та хвилинні випадки. Запропоновано багато принципів проектування мережі для класифікації зображень. Те саме також є істотно корисним для розпізнавання об'єктів.

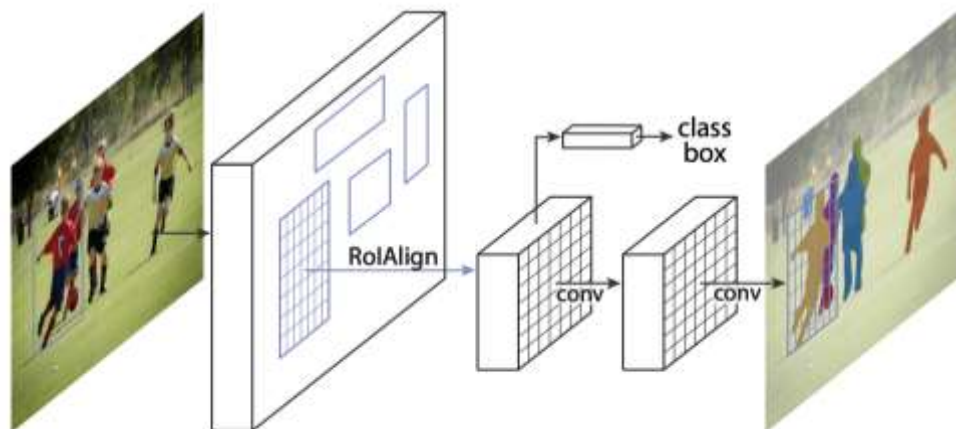


Рисунок 2.3 – Структура мережі Mask R-CNN

Важливим завданням у контексті сегментації зображень є надання можливості глибокій нейронній мережі усвідомлювати свою якість прогнозування [8, 9]. Для цілей сегментації екземпляру оцінка достовірності класифікації екземпляру використовувалася як оцінка якості маски більшістю підходів сегментації екземплярів. На противагу цьому підходу, якість маски, визначена кількісно, як перекривання над об'єднанням або IoU між передбаченою маскою екземпляра та основною істиною маскою, не була

належним чином пов'язана з класифікаційним балом. Для визначення перетину об'єднання використовують відношення, що зображено на рисунку 2.4, де Area of Overlap – площа перекривання передбачуваної та істинної масок об'єкта, Area of Union – площа об'єднання масок об'єкта.

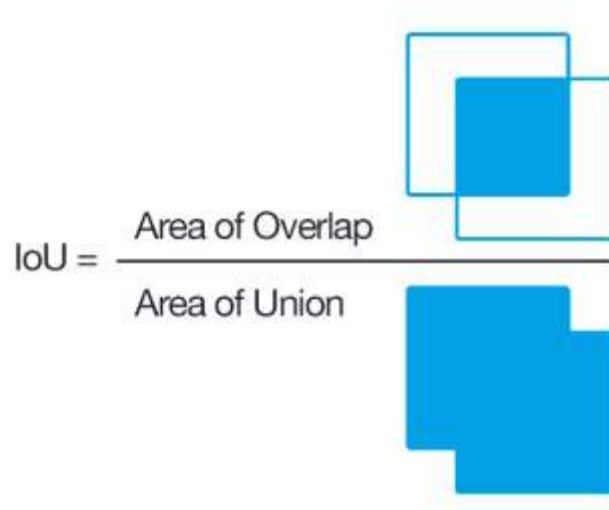


Рисунок 2.4 – Візуальне представлення відношення перекривання над об'єднанням

Дослідники вивчали цю проблему і згодом запропонували RCNN з оцінкою маски. Автори цієї роботи зазначають, що, не втрачаючи загальності, вони працювали над Mask R-CNN і додали додатковий головний модуль MaskIoU, який вивчає оцінку маски, сумісну з Mask-IoU. Вони стверджують, що їхня техніка концептуально проста. Mask R-CNN був поєднаний з MaskIoU надбудовою, що використовує характеристики екземпляру та передбачувану маску в поєднанні [19]. Ця схема використовується для прогнозування IoU між вхідною маскою та семантичною правдивою маскою. На рисунку 2.5 показана архітектура R-CNN з оцінкою маски. Модель містить мережевий блок, який має можливість дізнатися передбачувану якість маски. Цей мережевий блок, запропонований авторами, поєднує в собі особливості екземпляра з передбачуваною маскою, щоб додатково передбачати прогнозовану маску IoU. Цей підхід до оцінки маски обчислює помилку вирівнювання між якістю

маски та оцінкою маски, таким чином покращуючи ефективність завдання сегментації екземпляра, надаючи пріоритет кращим прогнозам масок при оцінці середнього передбачення (AP – average precision) COCO. Після великих експериментів із використанням набору даних COCO запропонований підхід послідовно та помітно вдосконалює різні моделі. Це навіть перевершує ефективний підхід Mask R-CNN [19].

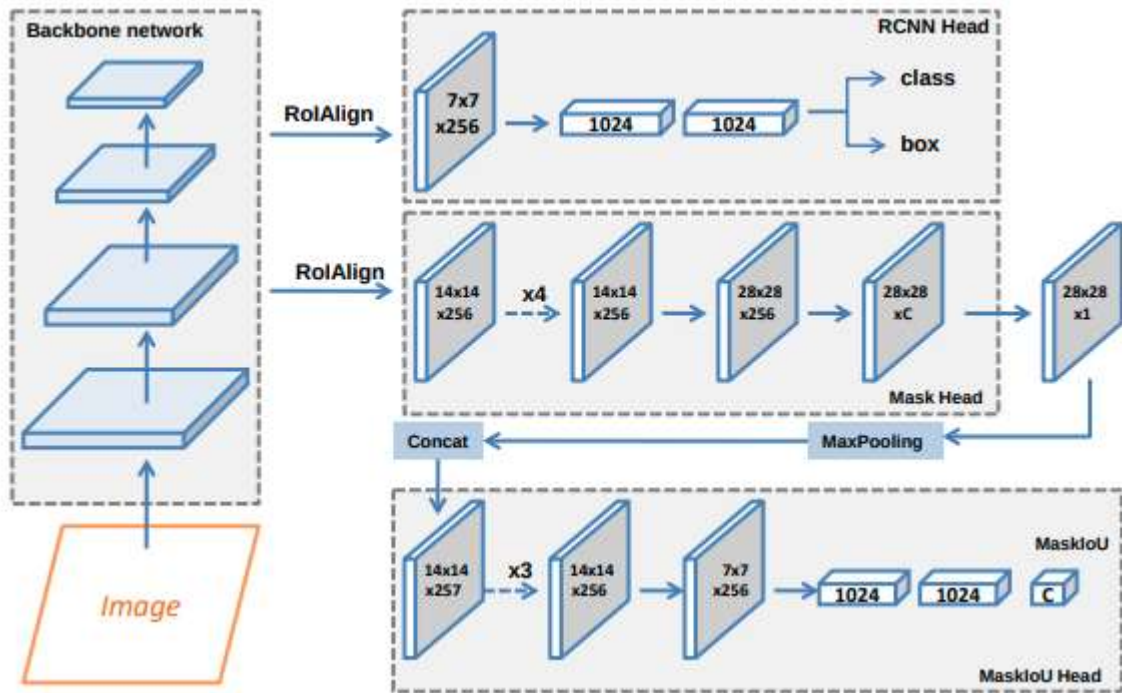


Рисунок 2.5 – Структура мережі Mask Scoring R-CNN

Дослідивши найкращі моделі сегментаційних нейронних мереж, прийнято рішення, використовувати для виявлення рекламних банерів саме сегментацію екземпляра та архітектуру Mask R-CNN, що реалізує даний тип сегментації. Оскільки передбачається створення власної надбудови до Mask R-CNN, то покращена мережа Mask Scoring R-CNN вимагає додаткових зусиль, щодо інтеграції власної надбудови з запропонованою.

2.2 Структура та функціонування згорткової нейронної мережі

Згорткові нейронні мережі витягують семантичну інформацію високого рівня за шаром із вхідного шару вихідних даних шляхом складання ряду операцій, таких як операція згортки, операція об'єднання та відображення нелінійних функцій активації. Цей процес називається операцією зворотного зв'язку. Різні типи операцій у згорткових нейронних мережах називаються шарами. Операції згортання – згорткові шари, а операції об'єднання – об'єднання шарів. Останній шар згорткової нейронної мережі перетворює цільові завдання (класифікація, регресія, тощо) у цільову функцію. Обчислюючи похибку або втрату між передбачуваним значенням і реальним значенням, шар за шаром за допомогою алгоритму зворотного поширення помилки для оновлення параметрів кожного шару, а потім назад і знову, поки мережева модель сходиться [8, 15].

Згорткова мережа приймає рівень операції згортки як основну одиницю і розміщує його на вихідних даних по одному шару за кожним попереднім і закінчує функцією втрат. Дані в кожному шарі є тривимірним тензором. У програмах комп'ютерного зору рівень даних згорткової нейронної мережі, як правило, є зображенням кольорового простору RGB: три канали (R, G, B), відповідають червоному, зеленому та синьому кольору. Нарешті, вся мережа закінчується обчисленням функції втрат [8].

Маска R-CNN концептуально проста: Faster R-CNN має два виходи для кожного об'єкта-кандидата, мітку класу та зміщення обмежувальної рамки; до цього ми додаємо третю гілку, яка виводить маску об'єкта. Таким чином, Mask R-CNN є природною та інтуїтивно зрозумілою ідеєю. Але додатковий вивід маски відрізняється від виходів класу та вікна, що вимагає вилучення набагато більш тонких просторових макетів об'єкта. Далі ми представляємо ключові елементи Mask R-CNN, включаючи вирівнювання між пікселями, яке є основним відсутнім фрагментом Fast / Faster R-CNN.

Ми починаємо з короткого огляду Faster R-CNN детектора. Faster R-CNN складається з двох етапів. На першому етапі, який називається мережа регіональних пропозицій (RPN), пропонуються обмежувальні рамки об'єктів-кандидатів. Другий етап, Fast R-CNN, витягує функції за допомогою RoIPool з кожного поля регіональних кандидатів та виконує класифікацію й регресію обмежувальної рамки. Функції, що використовуються на обох етапах, можуть бути спільними для швидшого опрацювання [18].

Mask R-CNN приймає ту саму двоступеневу процедуру, з однаковим першим етапом (RPN). На другому етапі, паралельно з прогнозуванням зміщення класу та рамки, Mask R-CNN також видає двійкову маску для кожного регіону інтересу (RoI). Даний підхід наслідує Faster R-CNN, який паралельно застосовує класифікацію та регресію обмежувальної рамки. Формально, під час навчання, ми визначаємо похибку декількох процесів на кожному вибірковому RoI як $L = L_{cls} + L_{box} + L_{mask}$, де L_{cls} – похибка класифікатора, L_{box} – похибка обмежувальної рамки об'єкта та L_{mask} – похибка сегментованої маски [17, 20].

Гілка маски має розмірний вивід Km^2 для кожного RoI, який кодує K бінарні маски з роздільною здатністю $m \times m$, по одній для кожного з класів K . Для RoI, пов'язаного з класом k , L_{mask} визначається лише на k -ій масці (інші виходи маски не сприяють втраті). Наше визначення L_{mask} дозволяє мережі створювати маски для кожного класу без конкуренції між класами; ми покладаємось на спеціальну гілку класифікації, щоб передбачити мітку класу, що використовується для вибору вихідної маски. Це відокремлює маску та передбачення класу. Це відрізняється від загальноприйнятої практики при застосуванні FCN до семантичної сегментації, яка зазвичай використовує софтмакс на піксель та мультиноміальну перехресну ентропію. У цьому випадку маски в різних класах змагаються. Експериментально ми показуємо, що це формулювання є ключовим для хороших результатів сегментації екземплярів [20].

Маска кодує просторовий макет вхідного об'єкта. Таким чином, на відміну від міток класів або зсувів рамок, які неминуче згортаються на короткі вихідні вектори повно-зв'язними шарами, витяг просторової структури масок може бути вирішений природним шляхом, використовуючи відповідність між вхідними та вихідними пікселями, що забезпечується згортками.

Зокрема, ми передбачаємо маску $m \times m$ від кожної RoI за допомогою FCN. Це дозволяє кожному шару у гілці маски підтримувати явний просторовий макет об'єкта $m \times m$, не згортаючи його у векторне представлення, у якому відсутні просторові розміри. На відміну від попередніх методів, які вдаються до повно-зв'язних шарів для прогнозування маски, згорткове подання вимагає меншої кількості параметрів і є більш точним. Ця поведінка відповідності пікселів вимагає, щоб функції RoI, які самі по собі є невеликими картами об'єктів, були добре вирівняні, щоб точно зберігати явну просторову відповідність за пікселем. Це спонукало дослідників розробити наступний рівень RoIAlign, який відіграє ключову роль у прогнозуванні маски [18, 19, 20].

Вирівнювання регіону інтересів або RoIAlign – це операція з вилучення невеликої карти об'єктів з кожного RoI у завданнях на основі виявлення та сегментації. Він усуває жорстке квантування пулу RoI, правильно вирівнюючи витягнуті функції з вхідними даними. Щоб уникнути будь-якого квантування меж або зон RoI, RoIAlign використовує дволінійну інтерполяцію для обчислення точних значень вхідних характеристик у чотирьох регулярно відібраних місцях у кожній зоні RoI, а результат потім агрегується (використовуючи максимальне або середнє) [18].

Потім вихід з шару RoIAlign подається в головну маску, яка складається з двох згорткових шарів. Він генерує маску для кожного RoI, таким чином, сегментуючи зображення з пікселя в піксель [19].

Моделі сегментації можуть створювати надмірно плавні межі, які можуть бути неточними для об'єктів або сцен з нерегулярними межами. Для отримання чітких меж сегментації в якості вдосконалення до існуючої моделі

було додано модуль нейронної мережі точкового візуалізації, який називається PointRend. Цей модуль спирається на методологію з класичної комп'ютерної графіки та надає перспективу візуалізації до проблеми сегментації. Моделі сегментації зображень часто передбачають мітки на звичайній сітці з низькою роздільною здатністю, наприклад, $1/8$ вхідного сигналу. Ці моделі використовують інтерполяцію, щоб збільшити прогнози до вихідної роздільної здатності. На відміну від цього, PointRend використовує ітераційний алгоритм підрозділу для підвищення масштабу прогнозів шляхом передбачення міток точок у вибраних місцях навченою невеликою нейронною мережею. Цей метод дозволяє ефективно виводити дані з високою роздільною здатністю [15, 17].

Детальна архітектура мережі відображена на рисунку 2.6.

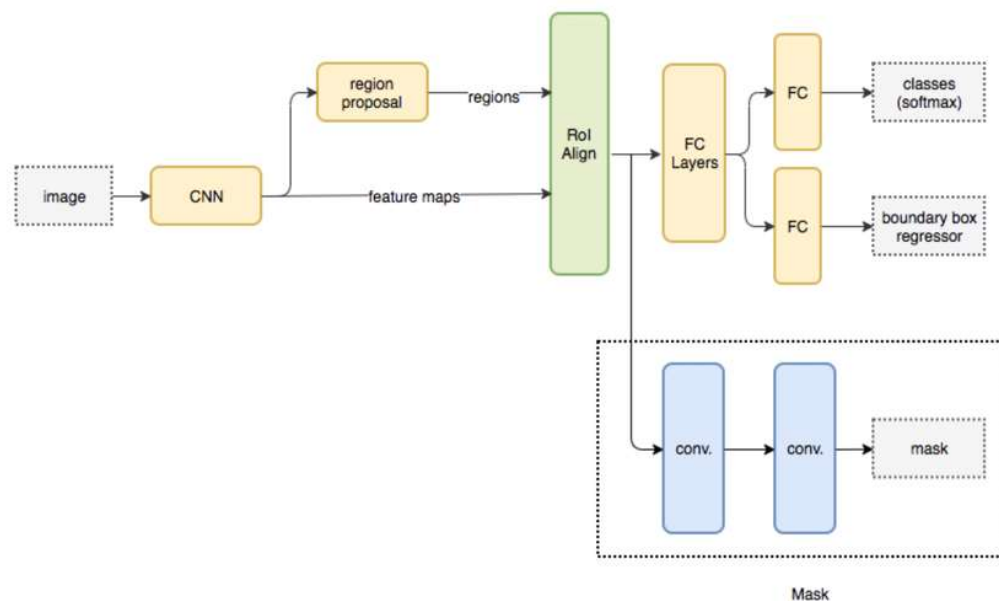


Рисунок 2.6 – Архітектура нейромережі Mask R-CNN

2.3 Особливості процесу виявлення рекламних банерів на відео

Мережа виконуватиме основну та найскладнішу частину інформаційної технології – виділяти сегментовані маски рекламних банерів на відеокадрах.

Основним ускладненням є додаткове визначення типу рекламного банера, враховуючи це в основу архітектури буде включено додатковий класифікатор.

Особливості реалізації нейромережі з додатковим класифікатором, зумовлені потребою в покращенні сегментації рекламних банерів різного типу, таких як: біл-борди, настінна реклама, футбольні банери. Даний підхід запропоновано використовувати спираючись на дослідження Mask Scoring R-CNN [19], через додаткові надбудови можна досягти кращих результатів сегментації зображення. Класифікатор типів реклами являє собою ще один повно зв'язний шар мережі, який визначає тип реклами якому належать функціональні характеристики й не потребує значних змін в архітектурі мережі.

З архітектурної сторони, функціонування нейронної мережі буде представлено у вигляді класу, об'єктно орієнтованого підходу проектування. Дане представлення об'єкту мережі дозволить відокремити її функціонал від головної частини програми та зберегти архітектуру – модель, представлення, конструктор (MVC). Такий підхід в свою чергу забезпечує відсутність залежності від внутрішнього стану інформаційної технології. Виконання дій над вхідними даними, буде реалізовано через методи класу, що відповідають за одну конкретну дію [20].

2.4 Формування навчального набору

Для забезпечення потенціалу мережі виявляти рекламні банери потрібно провести попереднє навчання. Нейронна мережа буде навчатись на власному наборі даних, що складається з зображень, які містять рекламні банери. Кожен елемент навчального набору містить 3 параметра, такі як: зображення, вихідна маска, клас об'єкта. Ці 3 характеристики подаються на вхід до нейронної мережі. На рисунку 2.7 можна наглядно побачити приклад елемента навчального набору, де зображено сам рекламний банер на футбольному полі та його маску. Зауважте, що представлення рекламного банеру відбувається

одразу в декількох перспективах, та з найповнішим представленням контексту, тобто використовуються потенційні сцени, що покращують здатність мережі до узагальнення. Рамка синього кольору, відображає межі рекламного банеру, а червонуватий колір – класове представлення кожного пікселя в межах банеру.



Рисунок 2.7 – Приклад візуальної частини вхідних даних неймережі

2.5 Розробка алгоритму заміни рекламних банерів на відео

Наступним етапом обробки буде визначення області, що придатна для вставки банеру. Дані обчислення будуть виконуватись виключно математичними формулами, що враховують перспективу банеру, кольорову гаму, реалістичність вставки та інші візуальні характеристики [21]. Зокрема, важливу роль відіграє згладжування кутових координат вставленого банера, без цієї операції об'єкт буде різко змінювати своє положення на декілька пікселів. Причиною цього є обмеженість точності нейронної мережі, що сегментує.

Для вбудовування в картинку, планується використовувати афінні перетворення, які враховують характеристики сегментованої картинки та переносять їх на цільовий об'єкт [21]. Афінне перетворення – відображення $f: R^n \rightarrow R$, можна записати у вигляді

$$f(x) = M \cdot x + v \quad (2.1)$$

Зазвичай для виконання таких перетворень виконуються маніпуляції над матрицями, оскільки саме зображення представлено у вигляді матриці – тензору в тривимірному просторі [21].

Згладжування буде виконуватись спеціальними фільтрами, які схожі зі згортковими функціями, проте з певним ступенем шуму. В нашому випадку гарним рішенням буде фільтр Савицького-Голея, який має просту реалізацію та чудову швидкодію [21].

$$Y_j = \sum_{i=\frac{1-m}{2}}^{\frac{m-1}{2}} C_i y_{j+i}, \text{ при чому } \frac{m-1}{2} \leq j \leq n - \frac{m-1}{2} \quad (2.2)$$

Дана функція згладжує контури банерів, для візуальної стабільності та реалістичності.

Об'єктно-орієнтована парадигма дозволяє відокремити важливі функціональні модулі як класи й зручно налаштувати взаємодію з ними. Основний клас буде опрацьовувати команди користувача, та делегувати задачі класам нижчого рівня. Вхідними даними до системи є відео файл, що завантажується користувачем. В результаті, система буде повертати на вихід вхідний відео-файл, проте зі зміненими рекламними банерами [22].

Етапи інформаційної технології проектованої інформаційної технології повинен забезпечувати таку послідовність дій:

1. Процес завантаження відео файлу користувачем
2. Процес вибір банерів представлених до заміни
3. Процес встановлення часових проміжків на яких буде відбуватись заміна
4. Процес опрацювання відео
5. Процес формування посилання, на завантаження опрацьованого відео.

Загальний алгоритм роботи інформаційної технології заміни рекламних банерів на відео, представлено на рисунку 2.8.

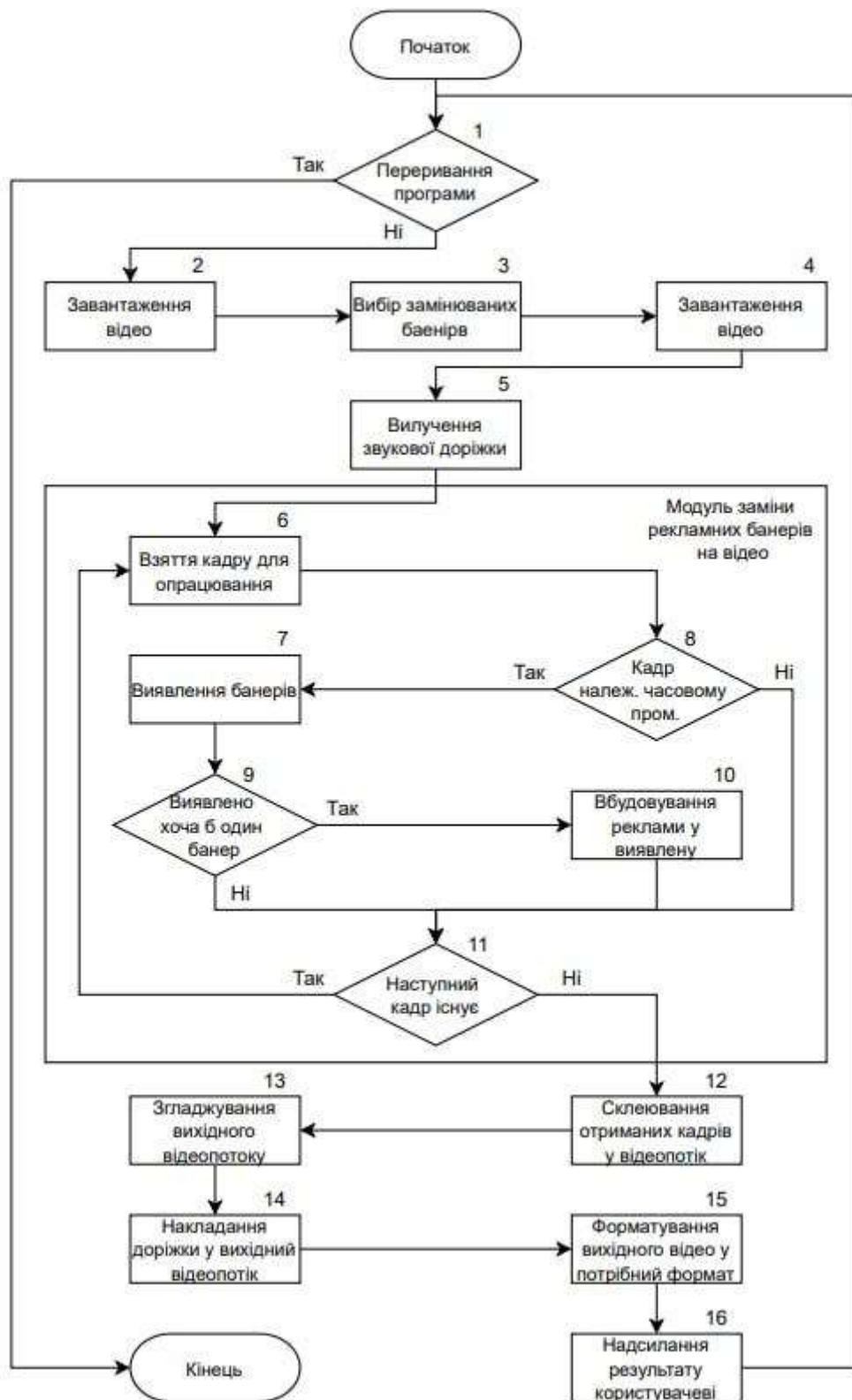


Рисунок 2.8 – Загальний алгоритм інформаційної технології заміни рекламних банерів на відео

2.6 Розробка UML-діаграми класів

Розробимо загальну структурну схему програми заміни рекламних банерів на відео. Програма складається з 4 основних компонентів:

- веб-клієнт;
- нейронна мережа Mask R-CNN
- програмний інтерфейс додатку(API);
- файлове сховище рекламних банерів доступних до заміни;

Опишемо кожен із компонентів.

Веб-клієнт представлятиме собою спрощену платформу для завантаження відео, де користувачі, можуть завантажувати відео та встановити відповідність рекламних банерів для заміни. Клієнт може перейти на веб сторінку застосунку та в результаті отримати відео з заміненими рекламними банерами.

Програмний інтерфейс представляє собою серверну частину, включає обробку даних, перевірку запитів, завантаження даних, завантаження замінюваної реклами та визначення часових проміжків де потрібно замінити рекламу.

Нейронна мережа виконує функцію пошуку банеру, де вхідне відео завантажувється через веб-інтерфейс, потім обираються види реклами, що будуть змінюватися. В результаті отримуємо підготовлений набір даних про місце вбудовування реклами.

Файлове сховище рекламних банерів, являє собою віддалений файловий сервер, що містить зразки рекламних банерів, які можуть бути замінені. Варто зауважити, що сервіс не має стану, отже після опрацювання відео, вся вхідна інформація видаляється, тому файли зберігаються тимчасово в фізичній пам'яті комп'ютера.



Рисунок 2.9 – Загальна структурна схема інформаційної технології заміни рекламних банерів у відео

Модель – це абстракція, будь-який образ, аналог якого-небудь об’єкту, процесу або явища [22, 23].

UML – уніфікована мова моделювання, використовується у парадигмі об’єктно-орієнтованого програмування. Є невід’ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем [22, 23, 24].

Для опису основного циклу роботи системи, можна використовувати різні діаграми, зокрема, діаграми класів та діаграми послідовностей [23].

Діаграма класів — статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. На рисунку 2.10 зображено діаграму класів інформаційної технології заміни рекламних банерів [24].

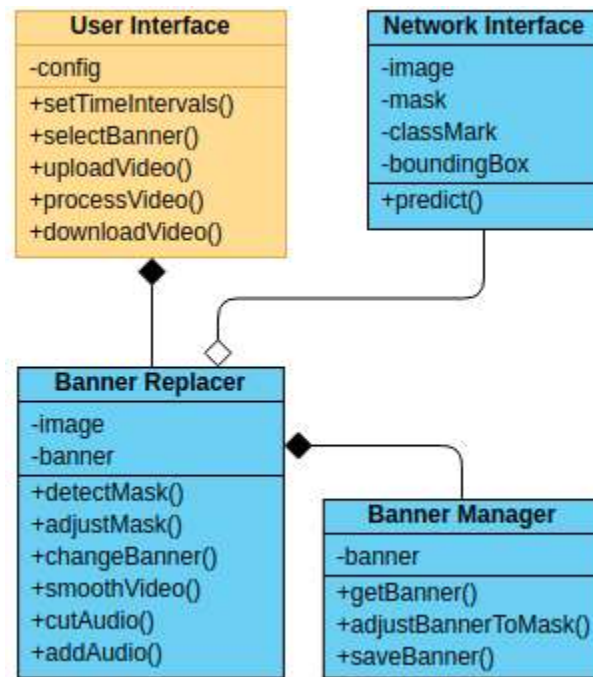


Рисунок 2.10 – Діаграма класів інформаційної технології заміни рекламних банерів

Діаграма послідовностей відноситься до діаграм взаємодії UML, що описує поведінкові аспекти системи, але розглядає взаємодію об'єктів в часі. Як і діаграма взаємодії, відображає тимчасові особливості передачі та прийому повідомлень об'єктами. Оскільки, діаграма послідовностей є розширеною версією діаграми взаємодії, то використаємо саме її для відображення об'єктів системи та їх взаємодії [24].

До об'єктів діаграми послідовностей віднесемо зазначені у попередньому розділі компоненти системи, а також додамо об'єкт користувача системи для відображення взаємодії. В результаті, отримали 5 об'єктів: веб-клієнт, програмний інтерфейс, нейронна мережа Mask R-CNN, файлове сховище даних, користувач. Розроблена діаграма послідовностей зображена на рисунку 2.11.

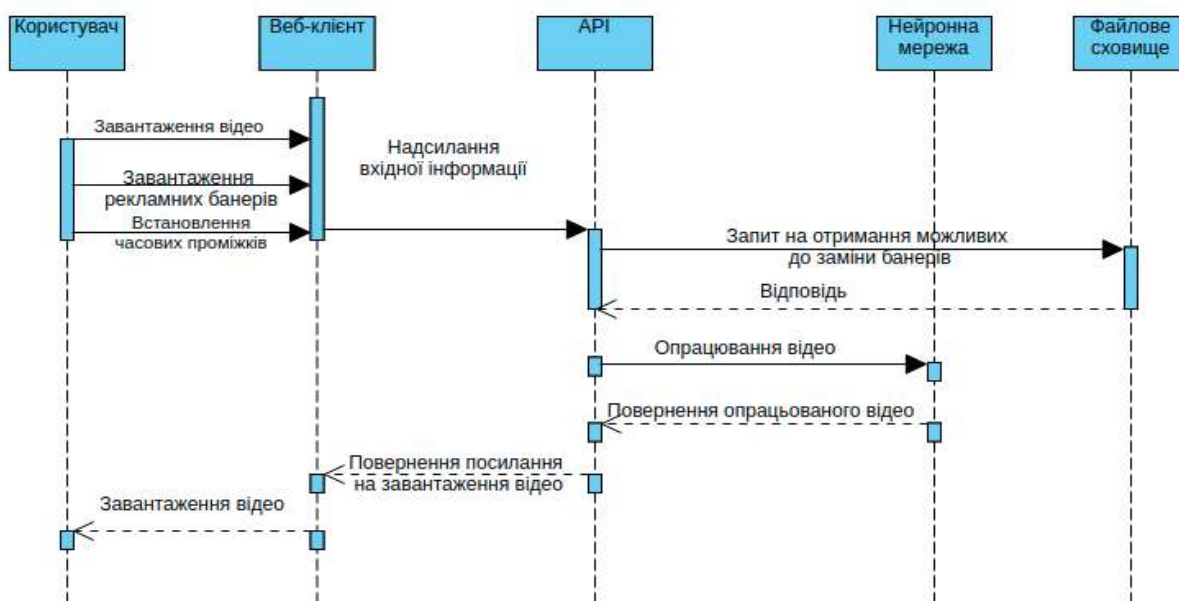


Рисунок 2.11 – Діаграма послідовностей інформаційної технології заміни рекламних банерів на відео

2.7 Висновок до розділу 2

У цьому розділі було проаналізовано різні структури нейронних мереж та обгрунтовано вибір нейронної мережі для даної задачі. У даному розділі було виділено основні компоненти інформаційної технології заміни рекламних банерів на відео та побудовано діаграму послідовності дій робочого циклу програми, що спрощує процес створення програмного забезпечення.

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ЗАМІНИ РЕКЛАМНИХ БАНЕРІВ НА ВІДЕО

3.1 Обґрунтування вибору мови та середовища програмування для реалізації програми заміни рекламних банерів на відео

Мова програмування – це набір лексичних абстракцій, що забезпечує зручний опис конкретних проблем, які сформовані людиною й потребують вирішення за допомогою комп'ютера. Мовою програмування пишеться програма, що дозволяє при її виконанні комп'ютером одержати конкретні результати. Мови програмування мають дві складові: синтаксис та семантика [25, 26]. Перехід від лексичних конструкцій до машинних кодів здійснює транслятор. Одна мова програмування може бути представлена різними реалізаціями, тобто зберігаються основні принципи й ідеї мови, розробляються зовсім різні транслятори, у синтаксисі допускаються деякі відмінності.

Визначимо інструменти, за допомогою яких буде створюватись програмна реалізація інформаційної технології заміни рекламних банерів у відео. Зокрема, необхідно визначити мову програмування для створення веб-клієнта, програмного інтерфейсу, нейронної мережі.

Для аналізу і порівняння було обрано 3 мови програмування: C#, JavaScript, Python.

C# – це мова програмування, що відноситься до мов із C-подібним синтаксисом. Ця мова ввібрала в себе елементи C++ і Java та була створена для розробки додатків на платформі Microsoft .NET Framework. Мова має строгу типізацію, проте має динамічну типізацію (тип змінної визначається в момент присвоєння їй значення, тобто тип змінної може бути різним в ході виконання програми). Використовується для створення клієнтських та веб-додатків. Крім того, в Visual Studio вбудований garbage-collector, який сам контролює пам'ять. C# використовується через свою універсальність [22].

JavaScript – мова програмування, що розвивається найбільш динамічно. Спочатку була орієнтована лише на веб-розробку, а саме на її клієнтську частину. Проте можливості мови розширюються з прийняттям нових стандартів EcmaScript, в тому числі і нові можливості об'єктно-орієнтованого програмування, що були недоступні раніше, наприклад, модифікатори доступу полів класів, можливість створення інтерфейсів, використання принципів поліморфізму та перевантаження методів. Так, на даний момент за допомогою JavaScript без труднощів розробляється і серверна частина веб-сайтів і додатків, а створення платформи Node.js дозволило створювати звичні програмні додатки [25].

Python є високорівневою мовою широкого застосування. Автор мови, Гвідо ван Россум, зробив неабиякий акцент на читабельності коду та синтаксисі. Мова дозволяє програмісту реалізувати задачі із значно меншою кількістю рядків коду. А також мова є багато-платформенною та застосовується для різних напрямків. Та найголовніше те, що мова підтримує популярні серед інших мов програмування парадигми: ООП, імперативне та функціональне програмування. Python є динамічно типізованим. Включає у себе автоматичне управління пам'яттю та надає масу додаткових бібліотек [26].

Всі три запропонованих мови, мають потужну підтримку пакетів машинного навчання, тому оцінювання та порівняння критеріїв, не буде враховувати даний пункт. Для оцінювання, буде враховуватись лише функціональні характеристики та можливості окремої мови.

У таблиці 3.1 наведена порівняльна характеристика мов програмування.

Таблиця 3.1 – Порівняльна характеристика мов програмування

Критерій/мова	C#	JavaScript	Python
Ручне управління пам'яттю	+	-	-
Необхідність компіляції	+	-	-
Необхідність у роботі з браузером	-	+	-
Виконання на клієнтській частині браузера	-	+	-
Пряме підключення скриптів	-	+	+
Використання на сервері	+	+	+
Наявність пакетів для машинного навчання	-	+	+

Для створення програмного інтерфейсу та розробки нейронної мережі було обрано мову Python, а для веб-клієнта - JavaScript. Python є основною мовою, що використовується для навчання та написання нейронних мереж, також він має чудові системні інструменти, що дозволяють отримати доступ до віддаленого файлового сховища банерів. Крім того, JavaScript є основною мовою, що використовується у розробці веб-застосунків, особливо в браузері, тому веб-клієнт буде написаний мовою JavaScript.

3.2 Вибір програмних інструментів

В якості інструмента для реалізації нейромережі вибрані наступні інструменти: NumPy, Pandas, TensorFlow.

NumPy це модуль з відкритим вихідним кодом для Python, який надає загальні математичні і числові операції у вигляді швидких функцій. Вони об'єднуються в високорівневі пакети. NumPy забезпечує функціонал, який можна порівняти з функціоналом MatLab та надає базові методи для маніпуляції з великими масивами і матрицями. SciPy (Scientific Python) розширює функціонал numpy величезною колекцією корисних алгоритмів,

таких як мінімізація, перетворення Фур'є, регресія та інші прикладні математичні техніки [26].

Pandas – це бібліотека Python, що надає широкі можливості для аналізу даних. Дані часто зберігаються в формі таблиць – наприклад, в форматах .csv, .tsv або .xlsx. За допомогою бібліотеки Pandas такі табличні дані дуже зручно завантажувати, обробляти і аналізувати за допомогою SQL-подібних запитів. А в зв'язці з бібліотеками Matplotlib і Seaborn – Pandas надає широкі можливості візуального аналізу табличних даних. Основними структурами даних в Pandas є класи Series і DataFrame. Перший з них являє собою одновимірний індексований масив даних деякого фіксованого типу. Другий – це двовимірна структура даних, що представляє собою таблицю, кожен стовпець якої містить дані одного типу. Можна представляти її як словник об'єктів типу Series. Структура DataFrame відмінно підходить для подання реальних даних: рядки відповідають ознаковим описам окремих об'єктів, а стовпці відповідають ознакам [26].

TensorFlow – це відкрита платформа для машинного навчання. Він має всеосяжну, гнучку екосистему інструментів, бібліотек та ресурсів спільноти, що дозволяє дослідникам впроваджувати найсучасніші технології ML, а розробники легко створюють та розгортають додатки, що працюють на ML. TensorFlow дозволяє легко створювати та навчати моделі машинного навчання, використовуючи інтуїтивно зрозумілі API високого рівня, такі як Keras, з швидким виконанням, що забезпечує негайну ітерацію моделі та легке налагодження. Проста та гнучка архітектура для швидкого перенесення нових ідей від концепції до коду, до найсучасніших моделей та публікації [26].

3.3 Навчання нейронної мережі

Метою даного підрозділу є дослідження та реалізація навчання нейронної мережі, тобто підбір та композиція найкращих оптимізаційних технік, щоб забезпечити найкращу точність передбачень нейронної мережі.

Для всіх експериментів використовується відтворена Mask R-CNN. Запропоновано використовувати пірамідальну мережу(FPN) на основі ResNet-18 для дослідження виокремлення та ResNet-18/50/101 на основі швидшої рекурентної, пірамідально та деформаційно-пірамідальної для порівняння того методу з іншими базовими результатами. Для ResNet-18 FPN розмір вхідних зображень змінено так, щоб мати 524 пікселів уздовж короткої осі та максимум 1024 пікселів вздовж довгої осі для навчання та тестування. Для ResNet-50/101 розмір вхідних зображень зменшено до 720 пікселів для короткої осі та 1440 пікселів для довгої осі навчання та тестування. Композиції мережі, що будуть використовуватись для навчання, представлені на рисунку 3.1. Тренування всіх мереж буде відбуватись протягом 100 епох, зменшуючи рівень навчання в 0,1 разів через 50 епох і 75 епох. В якості оптимізатора використовується синхронізований стохастичний градієнтний спуск(SGD) з імпульсом 0.9. На рисунку 3.2 можна помітити, що дві різних конфігурації мають різні показники навчання [9, 17, 18, 19].

Для тестування мережі буде використовуватись критерій не максимального стиснення(англ. non max suppression) і зберігається рейтинг 100 найкращих балів для кожного зображення. Результат навчання відображено на різних магістральних мережах, включаючи ResNet-18/50/101 та різну структуру, включаючи Faster R-CNN / FPN / DCN + FPN, щоб довести ефективність запропонованої мережі. Результати показані в таблиці 3.2. Використовується метрика середньої точності нашої мережі(APm) для звітування про результати сегментації екземплярів, а метрика середньої точності магістральної мережі(APb) для повідомлення про результати виявлення об'єктів. Як видно з таблиці, навчений Mask R-CNN для виявлення

банерів не чутливий до магістральної мережі і може досягти стабільного вдосконалення на всіх магістральних мережах: Навчений мною Mask R-CNN може отримати значне покращення (близько 1,5 AP). Спеціально для AP@0.75, запропонований метод може покращити базовий рівень приблизно на 2 бали. Таблиця 3.2 вказує на те, що наш Mask R-CNN стійкий до різних магістральних мереж, включаючи Fast R-CNN / FPN / DCN + FPN [21, 22].

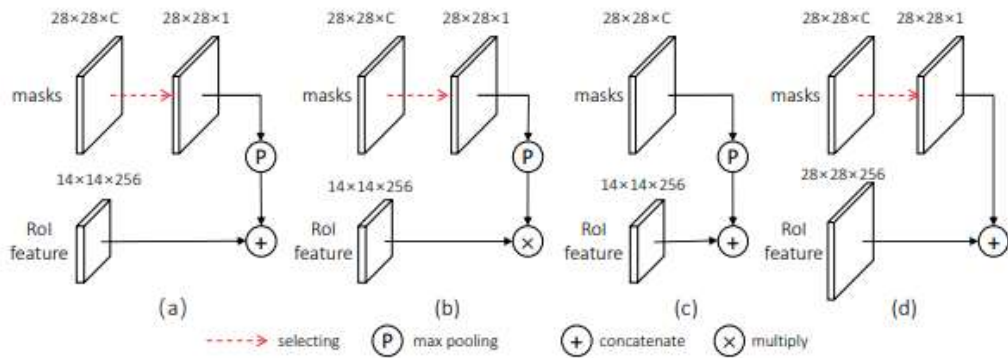


Рисунок 3.1 – Можливі надбудови до нейромережі Mask R-CNN

Порівняння різних складених конструкцій для сегментації рекламних банерів на відео наведено у таблиці 3.2.

Таблиця 3.2 – Порівняння різних методів об'єктної сегментації

Method	Backbone	AP	AP@0.5	AP@0.75	AP _S	AP _M	AP _L
MNC [7]	ResNet-101	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [23]	ResNet-101	29.2	49.5	-	-	-	-
FCIS+++ [23]	ResNet-101	33.6	54.5	-	-	-	-
Mask R-CNN [15]	ResNet-101	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN [15]	ResNet-101 FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN [15]	ResNeXt-101 FPN	37.1	60.0	39.4	16.9	39.9	53.5
MaskLab [3]	ResNet-101	35.4	57.4	37.4	16.9	38.3	49.2
MaskLab+ [3]	ResNet-101	37.3	59.8	36.6	19.1	40.5	50.6
MaskLab+ [3]	ResNet-101 (JET)	38.1	61.1	40.4	19.6	41.6	51.4
Mask R-CNN	ResNet-101	34.3	55.0	36.6	13.2	36.4	52.2
MS R-CNN		35.4	54.9	38.1	13.7	37.6	53.3
Mask R-CNN	ResNet-101 FPN	37.0	59.2	39.5	17.1	39.3	52.9
MS R-CNN		38.3	58.8	41.5	17.8	40.4	54.4
Mask R-CNN	ResNet-101 DCN+FPN	38.4	61.2	41.2	18.0	40.5	55.2
MS R-CNN		39.6	60.7	43.1	18.8	41.5	56.2

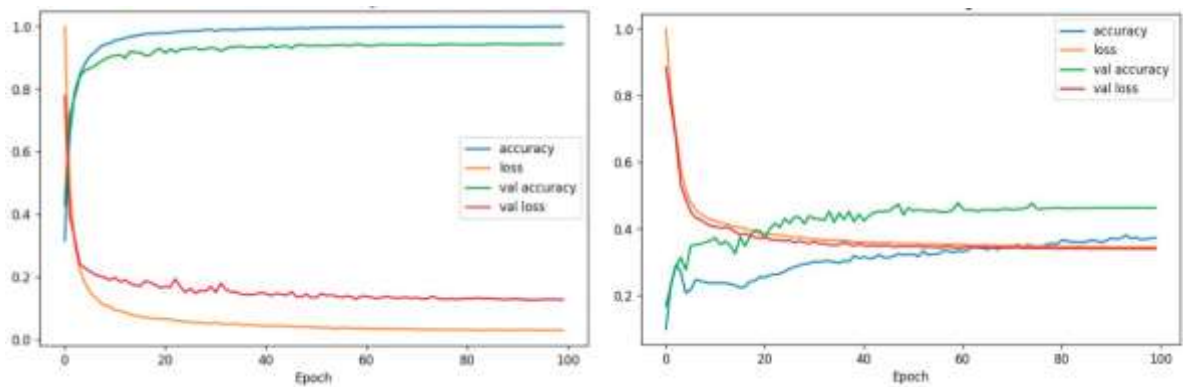


Рисунок 3.2 – Графіки навчання нейронної мережі

3.4 Програмна реалізація інформаційної технології заміни рекламних банерів на відео

Програмна реалізація передбачає створення веб інтерфейсу, що дозволяє користувачеві системи зручно взаємодіяти з інформаційною технологією заміни рекламних банерів на відео. Користувач через інтерфейс задає потрібну конфігурацію сервера та запускає опрацювання відео, після завершення цього етапу, надсилається посилання на завантаження відео [22].

Модуль виявлення рекламних банерів використовує заздалегідь навчену нейронну мережу, процес навчання якої описано в підрозділі 3.3. За одну ітерацію загального алгоритму, що зображений на рисунку 3.1 можна опрацювати лише 1 відео. Оскільки пріоритетним напрямком в даній роботі, є розробка інформаційної технології, то чимало уваги буде приділено інтерфейсу користувача та подальшому масштабуванню системи.

Ініціалізація програми буде виконуватись в контейнері віртуального оточення, що забезпечує стабільну роботу модуля, незалежно від оточення, тобто типу операційної системи, версій пакетів операційної системи та інших факторів. Для виконання ініціалізації програми використовується інструмент управління ізольованими контейнерами – Docker [27]. При чому контейнер використовує образ із вже встановленою в операційну систему бібліотекою Tensorflow, що має підтримку графічного ядра.

Вміст файлу конфігурації(Dockerfile) контейнера:

```
FROM tensorflow/tensorflow:2.2.0-gpu

RUN apt-get update
RUN apt-get install -y ffmpeg x264 libx264-dev
RUN apt-get install -y python3-dev
RUN apt-get install -y libsm6 libxext6 libxrender-dev

WORKDIR /usr/src/app

COPY requirements.txt .
RUN pip3 install -r requirements.txt
ENTRYPOINT [ "python3" ]
CMD ["app.py"]

COPY . .
```

Після ініціалізації контейнера, запускається сервер. Для написання серверної частини використано мікро-фреймворк Flask, що забезпечує можливість рендерингу веб сторінки на стороні сервера(server side rendering - SSR). Шаблони веб сторінок написані за допомогою мови гіпертекстової розмітки(HTML) та стилізовані каскадними таблицями стилів(CSS), а динамічних сценарії веб-сторінок реалізовано мовою JavaScript(JS)

Веб інтерфейс виконує свою основну роль, він є зручним та зрозумілим пересічному користувачеві. Фрагменти клієнтської частини програми, що використовують HTML, CSS та JS:

```
<style>
input[type=text] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  box-sizing: border-box;
}

#add, #remove, #final {
  background-color: #4CAF50; /* Green */
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
```



```

    text-decoration: none;
    display: inline-block;
    font-size: 16px;
}
</style>

</head>

<script>

    var index = 1
    function adds() {
        index++
        let form = document.getElementById("periods");
        var br = document.createElement("br");
        form.appendChild(br);
        let h3 = document.createElement("h3");
        h3.textContent = "Period " + index
        form.appendChild(h3)
        let start = document.createElement("INPUT")
        let finish = document.createElement("INPUT")
        let add = document.getElementById("add")
        let final = document.getElementById("final")
        add.parentNode.removeChild(add)
        final.parentNode.removeChild(final)
        start.setAttribute("type", "text")
        finish.setAttribute("type", "text")
        start.setAttribute("name", "start_" + index)
        finish.setAttribute("name", "finish_" + index)
        form.appendChild(start)
        form.appendChild(finish)
        form.appendChild(add)
        form.appendChild(final)
    }
</script>

<body>
<form action="/periods" method="post" id="periods">
    <h3>Period 1</h3>
    <input type="text" name="start_1">
    <input type="text" name="finish_1">
    <input type="button" value="Add period" onclick="adds();"
id="add">
    <input type='submit' value='Send' id="final">
</form>
</body>

```

Фрагменти коду серверної частини написані мовою Python з використанням фреймворку Flask:

```

@app.before_request
def before_request():

    with open(app.config["CONFIG_PATH"], 'r') as file:
        model_parameters = yaml.load(file, Loader=yaml.FullLoader)

        model_parameters["mask_path"] = app.config["MASK_PATH"]
        model_parameters["model_weights_path"] =
app.config["WEIGHT_FOLDER"] + '/mrcnn.h5'

    with open(app.config["CONFIG_PATH"], 'w') as write_file:
        documents = yaml.dump(model_parameters, write_file)

@app.route('/')
def init():
    return redirect('/set_video')

@app.route('/periods', methods=["POST", "GET"])
def set_time_periods():
    if request.method == "GET":
        return render_template('periods.html')
    else:

        data = request.form

        periods = defaultdict(dict)

        for timepoint in data:
            value = data[timepoint]
            point, n = timepoint.split('_')
            period = f"period_{n}"
            if value:
                periods[period][point] = value
            else:
                if period in periods:
                    del periods[period]

        periods = dict(periods)

        with open(app.config["CONFIG_PATH"], 'r') as file:
            model_parameters = yaml.load(file, Loader=yaml.FullLoader)

        model_parameters['periods'] = {}
        if periods:
            for period in periods:
                model_parameters['periods'][period] = {}
                start, finish = periods[period].values()
                model_parameters['periods'][period]['start'] =
convert_time(start)

```

```

        model_parameters['periods'][period]['finish'] =
convert_time(finish)

        with open(app.config["CONFIG_PATH"], 'w') as write_file:
            documents = yaml.dump(model_parameters, write_file)

        return render_template('banners.html')

```

Ключовий елемент інформаційної технології – програма для заміни рекламних банерів на відео, реалізований з використанням технологій визначених в розділі 3.2, як вже вище зазначалось на основі ООП парадигми. Головний модуль програми реалізовано як клас, щоб дати можливість створювати незалежні екземпляри, які будуть виконувати опрацювання, для кожного окремого користувача й мати власні змінювані параметри [22]. Ініціалізації нейронної мережі та завантаження ваг мережі відбувається при ініціалізації модуля в конструкторі класу. Фрагмент коду модуля:

```

class MRCNNLogoInsertion:

    def __init__(self):
        self.model = None
        self.frame = None
        self.num_detections = 0
        self.num_insertions = 0
        self.frame_num = 0
        self.load_smooth = True
        self.load_smooth_mask = True
        self.detection_successful = False
        self.corners = None
        self.replace = None
        self.fps = None
        self.key = None
        self.start = None
        self.finish = None
        self.config = None
        self.process = False
        self.to_replace = None
        self.point_ids = list()
        self.class_match = defaultdict(list)
        self.before_smoothing = True
        self.mask_id = None
        self.class_ids = list()
        self.backgrounds = {}
        self.masks_path = None
        self.cascade_mask = defaultdict(dict)

```

```

        self.saved_points = pd.DataFrame(columns=['x_top_left',
'y_top_left', 'x_top_right', 'y_top_right',
                                                'x_bot_left',
'y_bot_left', 'x_bot_right', 'y_bot_right'])

    def init_params(self, params):

        with open(params) as f:
            self.config = yaml.load(f, Loader=yaml.FullLoader)

        self.replace = self.config['replace']
        self.to_replace = list(self.replace.keys())
        self.masks_path = self.config['mask_path']

        if bool(self.config['periods']):
            self.key = list(self.config['periods'].keys())[0]
            self.start =
int(self.config['periods'][self.key]['start'])
            self.finish =
int(self.config['periods'][self.key]['finish'])
        else:
            self.process = True

        print("FPS: ", self.fps)
        print(f"Time periods for processing:
{self.config['periods']}")

    def detect_banner(self, frame):

        self.frame = frame

        self.__valid_time()
        if self.process:
            if self.before_smoothing:
                self.__detect_mask()
            else:
                if self.frame_num in self.class_match:
                    self.detection_successful = True
                else:
                    self.detection_successful = False
        self.frame_num += 1

    def __detect_mask(self):
        rgb_frame = np.flip(self.frame, 2)
        result = self.model.detect([rgb_frame])[0]
        class_ids = result['class_ids']
        masks = result['masks']

        mask_id = 0

```

```

for i, class_id in enumerate(class_ids):
    if class_id in self.to_replace:
        mask = masks[:, :, i].astype(np.float32)

        # mask_output = process_mask(mask)
        mask = process_mask(mask)
        # if mask_output:
        if mask.any():
            # mask, mask_points = mask_output

            # self.mask_ids.append((self.frame_num, i))
            # self.saved_masks.loc[f"{self.frame_num}_{i}"] =
mask_points

            banner_mask = np.zeros_like(rgb_frame)
            points = np.where(mask == 1)
            banner_mask[points] = rgb_frame[points]

            contours = get_contours(banner_mask)

            tmp_mask_id = []
            for cnt in contours:
                if cv2.contourArea(cnt) >
np.product(mask.shape) * 0.0008:
                    rect = cv2.minAreaRect(cnt)
                    box = cv2.boxPoints(rect).astype(np.int)

                    cnt_corners = found_corners(box)

                    self.point_ids.append((self.frame_num,
mask_id))

self.saved_points.loc[f"{self.frame_num}_{mask_id}"] = cnt_corners
                    tmp_mask_id.append(mask_id)
                    mask_id += 1

            self.class_match[self.frame_num].append({i:
class_id})

            self.cascade_mask[self.frame_num][i] = tmp_mask_id

            # self.num_detections += 1
            # np.save(os.path.join(self.masks_path,
f'frame_{self.frame_num}_{i}.npy'), mask.astype(np.int8))

            filename = os.path.join(self.masks_path,
f'frame_{self.frame_num}_{i}.dat')
            ft = np.memmap(filename, dtype='int8', mode='w+',
shape=mask.shape)
            ft[:] = mask[:]
            del ft

```

Інтерфейс користувача являє собою 5 веб сторінок, як є етапами загального алгоритму, на рисунках 3.3 й 3.4 можна побачити етапи вибору часових інтервалів та завантаження банерів до системи відповідно.

The screenshot shows a web form with two sections, "Period 1" and "Period 2". Each section contains two light blue input fields for time intervals. Under "Period 1", the fields contain "00:00:39" and "00:00:46". Under "Period 2", the fields contain "00:03:40" and "00:03:57". At the bottom of the form, there are two green buttons: "Add period" and "Send".

Рисунок 3.3 – Веб сторінка, що відповідає за встановлення часових інтервалів опрацювання відео

The screenshot shows a web form for uploading banner images. It lists six categories: "gazprom", "heineken", "mastercard", "nissan", "pepsi", and "playstation". Each category has a "Choose File" button and a text field. "gazprom" has "task3.jpg" in its field. "heineken", "nissan", "pepsi", and "playstation" have "No file chosen" in their fields. "mastercard" has "photo_2021...12-53-25.jpg" in its field. At the bottom, there is a green "Send" button.

Рисунок 3.4 – Встановлення відповідності заміни бажаних рекламних банерів на задані

Після встановлення всіх бажаних конфігурації, можна почати опрацювання відео. Процес виконання займає певний час, тому після початку опрацювання, користувач чекає на відповідь сервера. Для початку обробки, достатньо надіслати POST запит на сервер з параметром {process: True}. На рисунку 3.5 зображено відповідну кнопку, яка забезпечує даний функціонал.

3.5 Висновок до розділу 3

У розділі було обґрунтовано вибір технологій та підходів програмної реалізації інформаційної технології заміни рекламних банерів на відео. У результаті для реалізації, обрано мову програмування Python для навчання та імплементації нейронної мережі та серверної частини системи. Для реалізації клієнтської частини, обрано мову програмування JavaScript та інструменти розмітки інтерфейсу користувача HTML, CSS. Розроблено програму, що виконує заміну рекламних банерів на відео. Під час виконання даного розділу було реалізовано програмне рішення та наведено фрагменти коду й продемонстровано вигляд інтерфейсу користувача.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РОБОТИ ПРОГРАМИ ЗАМІНИ РЕКЛАМНИХ БАНЕРІВ НА ВІДЕО

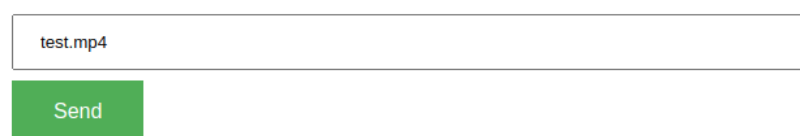
4.1 Тестування роботи інформаційної технології заміни рекламних банерів на відео

Тестування функціональності буде проводитись за такими критеріями:

- всі сторінки інтерфейсу користувача відображаються коректно, та виконують свій функціонал;
- при початку опрацюванні відео, сервер не припиняє роботу через довгий час очікування відповіді;
- після опрацювання користувач отримав посилання на відео;
- відео опрацьовано та реклама встановлена у потрібні місця;

Перейшовши за посиланням <http://0.0.0.0:5089/> на локальному комп'ютері, користувач отримує сторінку завантаження відео на сервер яка представлена на рисунку 4.1.

Upload video



test.mp4

Send

Рисунок 4.1 – Початкова сторінка веб-сервісу

Після завантаження відео та встановлення конфігурації запускається обробка відео. В процесі обробки виконується пошук місць для потенційної заміни банерів, валідація запропонованого місця та саме монтування банеру.

Після опрацювання, відео було завантажено на персональний комп'ютер користувача. Вихідне відео має складену назву з префіксом “processed_” та

повної назви відео, в тестовому випадку, отримали назву “processed_test.mp4” з вхідного відео під назвою “test.mp4”.

Пошук потенційних місць виконується нейронною мережею, для візуального відображення якості виявлення та сегментації рекламних банерів, взято уривок з футбольного матчу. Через велику кількість рекламних банерів на футбольних матчах, можна наявно продемонструвати точність виявлення різних типів банерів та компаній. На рисунку 4.2 зображено приклад виявлення рекламних банерів трьох компаній з середньою точністю 99,9%, така точність досягається шляхом попередньої класифікації типу банера.



Рисунок 4.2 – Приклад роботи нейронної мережі для виявлення рекламних банерів

Валідація виконується для забезпечення стабільної та нативної реклами в важкодоступні місця. Наприклад встановлення контурів банеру в місцях перекриття. На рисунку 4.3 зображено перекривання футбольного банеру суддею.

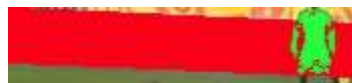


Рисунок 4.3 – Перевірка банеру на придатність заміни

Оскільки програма повертає вже готове відео, то будемо використовувати візуальне сприйняття для тестування якості заміни банерів. Для тестування якості заміни банерів було використано 2 відео. На першому

відео знаходяться уривки футбольних матчів, що містять рекламні банери спонсорів матчу, при чому заміна ускладнюється каскадами рекламних банерів одного типу. В свою чергу, другим відео є уривок кінофільму супермен, де виконується заміна настінних постерів на бажані рекламні інтеграції.

Рисунок 4.4 відображає виконання заміни в першому відео сегментованого рекламного банеру платіжної системи «visa» на банер компанії «1xbet». Заміна виглядає доволі реалістичною та майже непомітною для непроінформованого глядача.

На рисунку 4.5 представлено результат вбудовування інформаційного стенду компанії 1xbet в настінну рамку у другому відео.



Рисунок 4.4 – Результат заміни футбольного рекламного банеру



Рисунок 4.5 – Результат вбудовування інформаційного постера в настінну рамку

4.2 Порівняння з аналогами

Для порівняння результатів достовірності інформаційної технології заміни рекламних банерів на відео було обрано аналоги “toast” та “Amazon Recognition”. Варто зауважити, що Amazon Recognition є платформою яка дозволяє будувати власні рішення для вирішення потрібних задач, тому точність отриманої конфігурації може бути не максимальною.

Тестування кількісних характеристик відбувається на основі опрацьованих вихідних масках зображення, що придатні для заміни. Для тестування буде використовуватись та ж метрика що і для навчання мережі – перетин над об’єднанням. Порогом узгодження банеру буде коефіцієнт 0.9. У випадках коли один з аналогів, взагалі не знаходить потрібний банер, тобто міра перетину над об’єднанням менше порогового значення, присвоюється мітка успішності 0. У схожому випадку, коли мережа виявляє хибний банер, якого не існує в місці детекції, присвоюється та ж мітка [28].

У нашому випадку буде використовуватись метрика повноти (*recall*). В розпізнаванні образів, інформаційному пошуку та класифікації, влучність (англ. *precision*) є часткою релевантних зразків серед знайдених, тоді як повнота є часткою загального числа позитивних зразків, яку було дійсно знайдено. Як влучність, так і повнота, відтак ґрунтуються на розумінні та мірі релевантності. Влучність не слід плутати з точністю (англ. *accuracy*), яка є часткою правильно спрогнозованих результатів, як позитивних, так і негативних [28]. Формула метрики повноти має вигляд:

$$Recall = \frac{TP}{TP+FN} \quad (4.1)$$

де *TP* – кількість правильно визначених банерів, *FN* – кількість хибно незнайдених банерів.

Результат тестування аналогів зображено в таблиці 4.1.

Таблиця 4.1 – Порівняння достовірності розробленої програми заміни рекламних банерів на відео та аналогів

Достовірність заміни рекламних банерів у відсотках			
Обсяг тестової вибірки	250	250	250
Обсяг банеру в кадрі	Розроблена програма	«toast»	«Amazon Recognition»
Загальна площа фрагментів більше рівне 10%	72.07	69.40	64.2
Загальна площа фрагментів менше 10% й більше 5%	65.81	63.12	64.01
Загальна площа фрагментів менше 5%	66.3	58.97	62.76
4 і більше фрагменти	65.05	65.49	61.91
2-3 фрагменти	68.96	64.32	65.84
1 фрагмент	69.2	61.88	65.34
Усереднене значення	67.9	63.86	64.02

Можна помітити, що кожне з запропонованих рішень має свої переваги на зображеннях різного масштабу та різною часткою силуету банера в кадрі.

Розроблена програма має найкращу достовірність серед аналогів на зображеннях де банери видно чітко та їх площа займає до 10 відсотків від площі кадру. Аналог «toast» продемонстрував найкращу точність на кадрах, де площа банера більше ніж 10%. Рішення розроблене на платформі «Amazon Recognition» показує, що частка площі банера мало впливає на якість роботи аналогу.

Зокрема, результати тестування показали, що різна кількість банерів на зображенні не сильно впливає на точність програми.

4.3 Висновок до розділу 4

Аналіз результатів тестування показує, що усереднена достовірність заміни рекламних банерів є вищою за достовірність аналогів. Показники тестування на різних вибірках в більшості випадків, переважають достовірність аналогів. З таблиці 4.1 особливо помітно найвищу точність в категорії дрібних об'єктів, або об'єктів з частковою видимістю. Тобто, мета магістерської кваліфікаційної роботи досягнута – достовірність заміни рекламних банерів підвищена на 4%

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту інформаційної технології заміни рекламних банерів на відео. Особливістю розробки є підвищення достовірності виявлення та заміни рекламних банерів на відео за допомогою автоматизованої системи з використанням нейронних мереж та штучного інтелекту.

Аналогом може бути Toast, ціна 10 000 грн.; AmazonRecognition, ціна від 10 000грн до 100 000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

Продовження табл. 5.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 5.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ШБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	3	3	4
Цінова політика	3	4	3
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	4
Ринок збуту	4	3	4
Конкурентоспроможність	4	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	4
Матеріально-технічна база	4	3	4
Термін реалізації ідеї	4	4	4
Супровідна документація	4	4	4
Сума	44	42	46
Середньоарифметична сума балів	$(44+42+46) / 3 = 44$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що програмний продукт відрізняється від існуючих тим, що дана технологія є інформаційною технологією заміни рекламних банерів у відео, що відрізняється від існуючих функціональними можливостями, що забезпечило покращення процесу заміни рекламних банерів.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де M – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів в місяці, 23 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.1.

Таблиця 5.1 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	25000	1086,96	31	33695,652
Інженер	20000	869,57	31	26956,522
Всього				60652,17

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

5.2.2 Додаткова заробітна плата розробників, які приймали участь в розробці обладнання.

Додаткова заробітна плата прийнято розраховувати як 12 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 12 \% / 100 \% \quad (5.2)$$

$$Z_d = (60652,17 \cdot 12 \% / 100 \%) = 7278,26 \text{ (грн.)}$$

5.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (5.3)$$

$$H_z = (60652,17 + 7278,26) \cdot 22 \% / 100 \% = 14944,70 \text{ (грн.)}$$

5.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{T} \cdot \frac{t_{вик}}{12} \text{ [грн.]} \quad (5.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 80000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,35 міс.

$$A_{обл} = \frac{80000}{2} \times \frac{1,35}{12} = 4492,754 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.2. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

$$A_{н.р.} = Ц_{н.р.} * H_a * \frac{t_{вик.}}{12}$$

(5.5)

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних

нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $V_{нем.ак.} = 10100$ грн.

Таблиця 5.2 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія	80000	2	1,35	4492,754
Офісне обладнання	20300	4	1,35	570,018
Приміщення	900000	20	1,35	5054,348
Всього				10117,12

5.2.6 Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot \Pi \cdot \Phi \cdot K_{\Pi}, \quad (5.6)$$

де V – вартість 1 кВт-години електроенергії для 1 класу підприємства, $V = 6,2$ грн./кВт;

Π – встановлена потужність обладнання, кВт. $\Pi = 0,6$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

K_{Π} – коефіцієнт використання потужності, $K_{\Pi} = 0,9$.

$$B_e = 0,9 \cdot 0,6 \cdot 8 \cdot 31 \cdot 6,2 = 830,304 \text{ (грн.)}$$

5.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{iv}}{100\%}, \quad (5.7)$$

де H_{iv} – норма нарахування за статтею «Інші витрати».

$$I_e = 60652,17 \cdot 140\% / 100\% = 84913,04 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.8)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 60652,17 \cdot 150\% / 100\% = 90978 \text{ (грн.)}$$

5.2.9 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{\text{заг}} = 60652,17 + 7278,26 + 14944,70 + 10117,12 + 10100 + 830,30 + 84913,04 + 90978 = 279813,86 \text{ грн.}$$

5.2.11 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються ZB , визначається за формулою:

$$ZB = \frac{B_{\text{заг}}}{\eta} \text{ (грн)}, \quad (5.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ZB = 279813,86 / 0,5 = 559628 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

- а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);
- в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;
- г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);

- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (5.10)$$

де $\pm\Delta\Pi_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_0 = \Pi_0 \pm \Delta\Pi_0$;

Π_0 – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

ρ – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 5500 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 250 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 2500 шт., протягом другого року – на 1000 шт., протягом третього року на 500 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*250 + (5500 + 250) * 2500) * 0,8333 * 0,32 * (1 - 0,18) = 3006666,546 \text{ грн.}$$

$$\Delta\Pi_2 = (0*250 + (5500 + 250) * (2500+1000)) * 0,8333 * 0,32 * (1 - 0,18) = 4400666,491 \text{ грн.}$$

$$\Delta\Pi_3 = (0*250 + (5500 + 250) * (2500+1000+500)) * 0,8333 * 0,32 * (1 - 0,18) = 5029333,132 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 12436666,17 грн.

5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned} \text{ПП} = & (3006666,546/(1+0,1)^1) + (4400666,491/(1+0,1)^2) + (5029333,132/(1+0,1)^3) = \\ & 2733333,22 + 3636914,455 + 3778612,421 = 10148860,1 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{inv} * ZB, \quad (5.12)$$

де k_{inv} – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{inv} = 2 \dots 5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 559628 = 1119255,43 \text{ грн.}$$

Тоді абсолютний економічний ефект E_{abs} або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV, \quad (5.13)$$

$$E_{abc} = 10148860,1 - 1119255,43 = 9029604,67 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (*IRR, Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_g . Для цього використаємо формулу:

$$E_g = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.14)$$

$T_{жс}$ – життєвий цикл наукової розробки, роки.

$$E_g = \sqrt[3]{(1 + 9029604,67/1119255,43) - 1} = 1,085$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_v > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_v}, \quad (5.16)$$

$$T_{ок} = 1 / 1,085 = 0,92 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,92 роки, то фінансування даної наукової розробки є доцільним.

5.4 Висновок до розділу 5

Економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 559628 гривень. Було прогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є конкурентоспроможним. Період окупності складе близько 0,92 роки.

ВИСНОВКИ

При виконанні магістерської кваліфікаційної роботи розв'язано задачу розробки інформаційної технології заміни рекламних банерів у відео за допомогою нейронних мереж та було запропоновано новий підхід до застосування рекламних інтеграцій у відео контент.

Всі завдання поставлені перед магістерською кваліфікаційною роботою виконані в повному об'ємі, а саме:

1. Проаналізовано існуючі рішення та відомі аналоги заміни рекламних банерів на відео.
2. Обґрунтовано вибір методу розв'язання задачі.
3. Обрано оптимальну архітектуру нейронної мережі та спроектовано структуру нейронної мережі для застосування.
4. Спроектовано інформаційну технологію заміни рекламних банерів на основі виявлених банерів нейронною мережею.
5. Реалізовано програму заміни рекламних банерів на відео.
6. Проведено тестування розробленої програми заміни рекламних банерів на відео.
7. Розроблено інструкцію користувача.

Інформаційна технологія реалізована з використанням згорткових нейронних мереж, що забезпечило досягнення визначеної мети дослідження. Реалізована модель дозволяє автоматизувати процес заміни рекламних банерів у відео та потребує лише втручання оператора для контролю якості та встановлення параметрів обробки .

Розроблену інформаційна технологія можливо покращити шляхом реалізації додаткового інтелектуального модуля, який зможе забезпечити попереднє опрацювання вхідних даних, для розподілення задач до під-модулів системи. Також значне покращення забезпечить інтеграція програмного

інтерфейсу з дистриб'юторами медіа контенту, даний крок дає можливість масштабувати підхід персоніфікації контенту.

Мета магістерської кваліфікаційної роботи досягнута – достовірність заміни рекламних банерів підвищена на 4%, метод виявлення та заміни рекламних банерів показує конкурентні результати серед аналогів. За рахунок додаткового шару нейронної мережі, вдалось покращити точність в певних випадках заміни.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Діденко Ю.В., Колесницький О.К. Молодь в науці, дослідження, проблеми й перспективи (МН-2023). Інформаційна технологія заміни рекламних банерів на відео. [Електронний ресурс] - <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12624>
2. Matthew Harrison, Julia Cupman, Oliver Truman, Paul Hague. Market Research in Practice: An Introduction to Gaining Greater Market Insight, 2016 [Електронний ресурс] – режим доступу: https://books.google.com.ua/books?hl=uk&lr=&id=XOunCwAAQBAJ&oi=fnd&pg=PR9&dq=advert+market+research&ots=W8IIRz3nYy&sig=V7HHTs0iqp5ndDTdUxJpNHSqwDg&redir_esc=y#v=onepage&q&f=false
3. Bo Pang, Kaiwen Zha, Yifan Zhang, Cewu Lu. Further Understanding Videos through Adverbs: A New Video Task [Електронний ресурс] – режим доступу: <https://ojs.aaai.org//index.php/AAAI/article/view/6855>
4. Michele Covell, Shumeet Baluja, Michael Fink. Advertisement Detection and Replacement using Acoustic and Visual Repetition [Електронний ресурс] – <https://storage.googleapis.com/pub-tools-public-publication-data/pdf/55.pdf>
5. Brian Wansink. COMPARISON ADVERTISING’S IMPACT ON CATEGORY SUBSTITUTION [Електронний ресурс] – режим доступу: DOI:10.2307/3151879
6. Perlich, C., Dalessandro, B., Raeder, T.. Machine learning for targeted display advertising: transfer learning in action. [Електронний ресурс] – Режим доступу: <https://doi.org/10.1007/s10994-013-5375-2>
7. ADVANCED ADVERTISING IS IMPERATIVE FOR BROADCASTERS LOOKING TO COMPETE WITH MAJOR INTERNET PLAYERS, ACCORDING TO REPORT [Електронний ресурс] – Режим доступу: <https://www.yospace.com/tag/dynamic-ad-replacement/>
8. ODSC - Open Data Science. Using the CNN Architecture in Image Processing [Електронний ресурс] – Режим доступу: <https://medium.com/@ODSC/using-the-cnn-architecture-in-image-processing-65b9eb032bdc>

9. V. P. Kozemiako ; O. K. Kolesnytskyj ; T. S. Lischenko ; W. Wojcik and A. Sulemenov Optoelectronic spiking neural network, Proc. SPIE 8698, Optical Fibers and Their Applications 2012, 86980M (January 11, 2013); <http://dx.doi.org/10.1117/12.2019340>
10. O. K. Kolesnytskyj, I. V. Bokotsey, S. S. Yaremchuk Optoelectronic Implementation of Pulsed Neurons and Neural Networks Using Bispin-Devices // Optical Memory & Neural Networks (Information Optics), 2010, Vol.19, №2, pp.154-165.
11. Soumyabrata Dev, Hossein Javidnia, Murhaf Hossari¹ , Matthew Nicholson, Killian McCabe, Atul Nautiyal, Clare Conran, Jian Tang, Wei Xu, Francois Pitie'. Identifying Candidate Spaces for Advert Implantation [Электронный ресурс] - Режим доступа: <https://arxiv.org/pdf/1910.03227.pdf>
12. E. Hoffmann. Standard Statistical Classifications: Basic Principles, 2009, ст. 215.
13. Transcreation & Advertising versioning, the cost effective way to update ads. Animated packshot replacement is a great way to avoid costly packshot shoots. [Электронный ресурс] - Режим доступа: <https://www.toasttv.co.uk/advertising-versioning/>
14. Ad Replacement Systems. [Электронный ресурс] – Режим доступа: <http://www.halfminute.co.uk/adreplacement.html>
15. Murhaf Hossari, Soumyabrata Dev , Matthew Nicholson, Killian McCabe, Atul Nautiyal, Clare Conran, Jian Tang, Wei Xu, Francois Pitie'. ADNet: A Deep Network for Detecting Adverts [Электронный ресурс] – Режим доступа: <https://arxiv.org/pdf/1811.04115.pdf>
16. Amazon Rekognition, Автоматизируйте анализ изображений и видео с помощью машинного обучения. [Электронный ресурс] – Режим доступа: <https://aws.amazon.com/ru/rekognition/>
17. Abdul Mueed Hafiz, Ghulam Mohiuddin Bhat. A Survey on Instance Segmentation: State of the art [Электронный ресурс] – Режим доступа: <https://arxiv.org/ftp/arxiv/papers/2007/2007.00047.pdf>
18. Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick. Mask R-CNN [Электронный ресурс] – Режим доступа: <https://arxiv.org/pdf/1703.06870.pdf>

19. Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, Xinggang Wang. Mask Scoring R-CNN [Електронний ресурс] – Режим доступу: <https://arxiv.org/pdf/1903.00241.pdf>
20. Understanding Region of Interest — (RoI Align and RoI Warp). [Електронний ресурс] – Режим доступу: <https://towardsdatascience.com/understanding-region-of-interest-part-2-roi-align-and-roi-warp-f795196fc193>
21. Richard Szeliski. Computer Vision Algorithms and Applications, 2020, ст. 170-189.
22. Maurizio Gabbrielli, Simone Martini. Programming Languages: Principles and Paradigms, 2013, ст. 74-91.
23. Загальна характеристика UML [Електронний ресурс] – Режим доступу: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=2>
24. Мартін Фаулер. UML. Короткий посібник по UML, 2008, ст. 70-73.
25. Д. Фленаган. JavaScript. Подробное руководство, 2016, ст. 14-23.
26. Jake VanderPlas. Python Data Science Handbook, 2017, ст. 9-12.
27. About Docker [Електронний ресурс] – Режим доступу: <https://www.docker.com/company>
28. Selecting Metrics for Machine Learning. [Електронний ресурс] – Режим доступу: <https://fayrix.com/blog/machine-learning-metrics>
29. Технологічний аудит [Електронний ресурс]. URL: <https://studfile.net/preview/7336428/>
30. Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської) та конструкторсько-технологічної роботи [Електронний ресурс]. URL: <https://web.posibnyku.vntu.edu.ua>.
31. Ефективність наукових досліджень [Електронний ресурс]. URL: <https://studfile.net/preview/6872280/page:25/>.

ДОДАТКИ

Додаток А (обов'язковий)

Результат перевірки на плагіат в онлайн-системі UNICHECK

Дата звіту:
19.12.2022 23:39:56 EET

ID користувача:
62038

Назва документа: 122МКР-ДіденкоЮВ2022

Кількість сторінок: 55 Кількість слів: 9823 Кількість символів: 75005 Розмір файлу: 981.00 KB ID файлу: 1013092359

9.66% Схожість

Найбільша схожість: 3.65% з джерелом з Бібліотеки (ID файлу: 1009723663)

0.38% Джерела з Інтернету

44

Сторінка 57

9.66% Джерела з Бібліотеки

95

Сторінка 57

0.23% Цитат

Цитати

1

Сторінка 58

Не знайдено жодних посилань

75.3% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 5%)

2.45% Вилучення з Інтернету

75

Сторінка 59

74.7% Вилученого тексту з Бібліотеки

101

Сторінка 59

Додаток Б (обов'язковий)

Лістинг програми

```

import sys
from threading import Thread
from numba import cuda
import tensorflow as tf
import gc

physical_devices = tf.config.experimental.list_physical_devices('GPU')
tf.config.experimental.set_memory_growth(physical_devices[0], True)

import glob

sys.path.append('../models/mrcnn')
from models.nn_models.MaskRCNN import myMaskRCNNConfig, MRCNNLogoInsertion
from models.nn_models.mrcnn import model as modellib
import cv2
from core.config import app
import os

os.environ["CUDA_VISIBLE_DEVICES"]="0"

class Compute(Thread):
    def __init__(self, request):
        Thread.__init__(self)
        self.request = request

    def run(self, config_file):
        status = process_video(config_file)
        print(status)
        self.restart_program()

    def restart_program(self):
        python = sys.executable
        os.execl(python, python, *sys.argv)

def add_audio(out_video_path):
    """
    Extract audio file from input video and add it to output video
    :param video_path: video path
    :return: output video name
    """
    video_name = out_video_path.split('/')[-1]
    audio_name = f"audio_{video_name.split('.')[0]}.mp3"
    input_video = os.path.join(app.config["UPLOAD_FOLDER"], video_name)
    output_audio = os.path.join(app.config["AUDIO_PATH"], audio_name)
    output_video = app.config["DOWNLOAD_FOLDER"] + '/sound_' + video_name
    os.system(f'ffmpeg -i {input_video} {output_audio}')
    if os.path.exists(output_audio):
        os.system(f'ffmpeg -i {out_video_path} -i {output_audio} -codec copy
        -shortest {output_video}')
        os.remove(out_video_path)
        os.remove(output_audio)

def process_video(config_file):

```

```

logo_inserter = MRCNNLogoInsertion()
logo_inserter.init_params(config_file)

config = myMaskRCNNConfig()
logo_inserter.model = modellib.MaskRCNN(mode="inference", config=config,
model_dir='/')

logo_inserter.model.load_weights(logo_inserter.config['model_weights_path'],
by_name=True)
source_link = logo_inserter.config['source_link']
saving_link = logo_inserter.config['saving_link']

print("Detection step")
cap = cv2.VideoCapture(source_link)
logo_inserter.fps = cap.get(cv2.CAP_PROP_FPS)
while cap.isOpened():
    ret, frame = cap.read()

    if ret:
        logo_inserter.detect_banner(frame)
    else: .reset()
gc.collect()

print('Insertion step')

logo_inserter.frame_num = 0
logo_inserter.before_smoothing = False
logo_inserter.init_params(config_file)

cap = cv2.VideoCapture(source_link)
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
four_cc = cv2.VideoWriter_fourcc('m', 'p', '4', 'v')
out = cv2.VideoWriter(saving_link, four_cc, logo_inserter.fps,
(frame_width, frame_height), True)

while cap.isOpened():
    ret, frame = cap.read()

    if cap.get(1) % 1000 == 0:
        print(f"Inserted
{round(cap.get(1)/cap.get(cv2.CAP_PROP_FRAME_COUNT) * 100, 3)}%")
        gc.collect()

        break

    if cap.get(1) % 1000 == 0:
        print(f"Processed
{round(cap.get(1)/cap.get(cv2.CAP_PROP_FRAME_COUNT) * 100, 3)}%")
        gc.collect()

cap.release()
device = cuda.get_current_device()
device

if cap.get(1) % 50 == 0:
    gc.collect()

if ret:
    logo_inserter.detect_banner(frame)
    logo_inserter.insert_logo()

    out.write(frame)
else:

```

```

        break

    cap.release()
    cv2.destroyAllWindows()
    out.release()

    add_audio(saving_link)

    files = glob.glob(app.config['MASK_PATH']+'/*.npy')
    for f in files:
        os.remove(f)

    return True

import numpy as np
import cv2

def found_corners(box):

    box.view('i8,i8').sort(order=['f0'], axis=0)

    left_side = box[:2]
    right_side = box[2:]

    left_bot_y = np.argmax(left_side, axis=0)[1]
    left_top_y = np.argmin(left_side, axis=0)[1]
    right_bot_y = np.argmax(right_side, axis=0)[1]
    right_top_y = np.argmin(right_side, axis=0)[1]

    top_left = left_side[left_top_y]
    bot_left = left_side[left_bot_y]
    top_right = right_side[right_top_y]
    bot_right = right_side[right_bot_y]

    return [*top_left, *top_right, *bot_left, *bot_right]

def get_contours(mask):

    small_kernel = np.ones((3, 3), np.uint8)
    kernel = np.ones((5, 5), np.uint8)

    gray_mask = cv2.cvtColor(mask, cv2.COLOR_RGB2GRAY)
    gray_erosion = cv2.erode(gray_mask, small_kernel, iterations=1)
    gray_dilation = cv2.dilate(gray_erosion, small_kernel, iterations=5)
    ret, thresh = cv2.threshold(gray_dilation, 127, 256, cv2.THRESH_BINARY)
    opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, small_kernel)
    gauss = cv2.GaussianBlur(opening, (5, 5), 1)
    dilation = cv2.dilate(gauss, small_kernel, iterations=1)
    erosion = cv2.erode(dilation, kernel, iterations=3)

    _, contours, _ = cv2.findContours(erosion, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    return contours

def create_background(logo_path, frame_shape):
    logo = cv2.imread(logo_path, cv2.IMREAD_UNCHANGED)

    if logo.shape[2] == 4:

```

```

        logo = cv2.cvtColor(logo, cv2.COLOR_BGRA2BGR)

    un, cnts = np.unique(logo, axis=1, return_counts=True)
    idx = cnts.argmax()
    pixel_value = logo[:, idx][0]
    background = np.full(frame_shape, pixel_value)

    return background

import numpy as np
import cv2
from scipy.signal import savgol_filter

def smooth_points(df):

    df.reset_index(drop=True, inplace=True)

    if df.shape[0] > 4:

        df['y_top_left'] = smooth_series(df['y_top_left'])
        df['y_top_right'] = smooth_series(df['y_top_right'])
        df['y_bot_left'] = smooth_series(df['y_bot_left'])
        df['y_bot_right'] = smooth_series(df['y_bot_right'])

        df['x_top_left'] = smooth_series(df['x_top_left'])
        df['x_top_right'] = smooth_series(df['x_top_right'])
        df['x_bot_left'] = smooth_series(df['x_bot_left'])
        df['x_bot_right'] = smooth_series(df['x_bot_right'])

    return df

def line_equation(left_point, right_point, new_point):
    x_left, y_left = left_point
    x_right, y_right = right_point
    return (new_point - x_left) * (y_right - y_left) / (x_right - x_left) +
    y_left

def process_mask(mask):
    # center_right = None
    # center_left = None

    mask_points = np.argwhere(mask == 1)
    if mask_points.any():

        max_x = mask_points[mask_points[:, 1].argmax()]
        min_x = mask_points[mask_points[:, 1].argmin()]
        top_y = mask_points[mask_points[:, 0].argmin()]

        if abs(max_x[0] - top_y[0]) > 10 and abs(min_x[0] - top_y[0]) > 10:

            left_y_shift, left_x_shift = np.abs(top_y - min_x)
            right_y_shift, right_x_shift = np.abs(top_y - max_x)

            left_angle = np.arctan(left_y_shift / left_x_shift) * 180 / np.pi
            right_angle = np.arctan(right_y_shift / right_x_shift) * 180 /
np.pi

            if left_angle < right_angle:
                mask[:, top_y[1]:] = 0

```

```
        else:
            mask[:, :top_y[1]] = 0

    return mask

def smooth_series(series):

    min_window = 3
    max_window = 33
    poly_degree = 2
    threshold = 10

    best_series = []

    # smoothing
    for wnd_size in range(min_window, max_window):
        if wnd_size > len(series):
            break
        if wnd_size % 2 == 0:
            continue
        new_series = savgol_filter(series, wnd_size, poly_degree)
        if max(abs(new_series - series)) < threshold:
            best_series = new_series

    return best_series
```


Додаток В (обов'язковий)**ІЛЮСТРАТИВНА ЧАСТИНА****ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЗАМІНИ РЕКЛАМНИХ БАНЕРІВ НА
ВІДЕО**

Виконав: студент 1-го курсу,
групи ІКН-21м
спеціальності 122 «Комп'ютерні науки»


(шифр і назва напрямку підготовки, спеціальності)



Діденко Ю.В.

(прізвище та ініціали)

Керівник: д.т.н., доц каф. КН



Колесницький О.К.

(прізвище та ініціал)

« 15 » 12

2022 р.

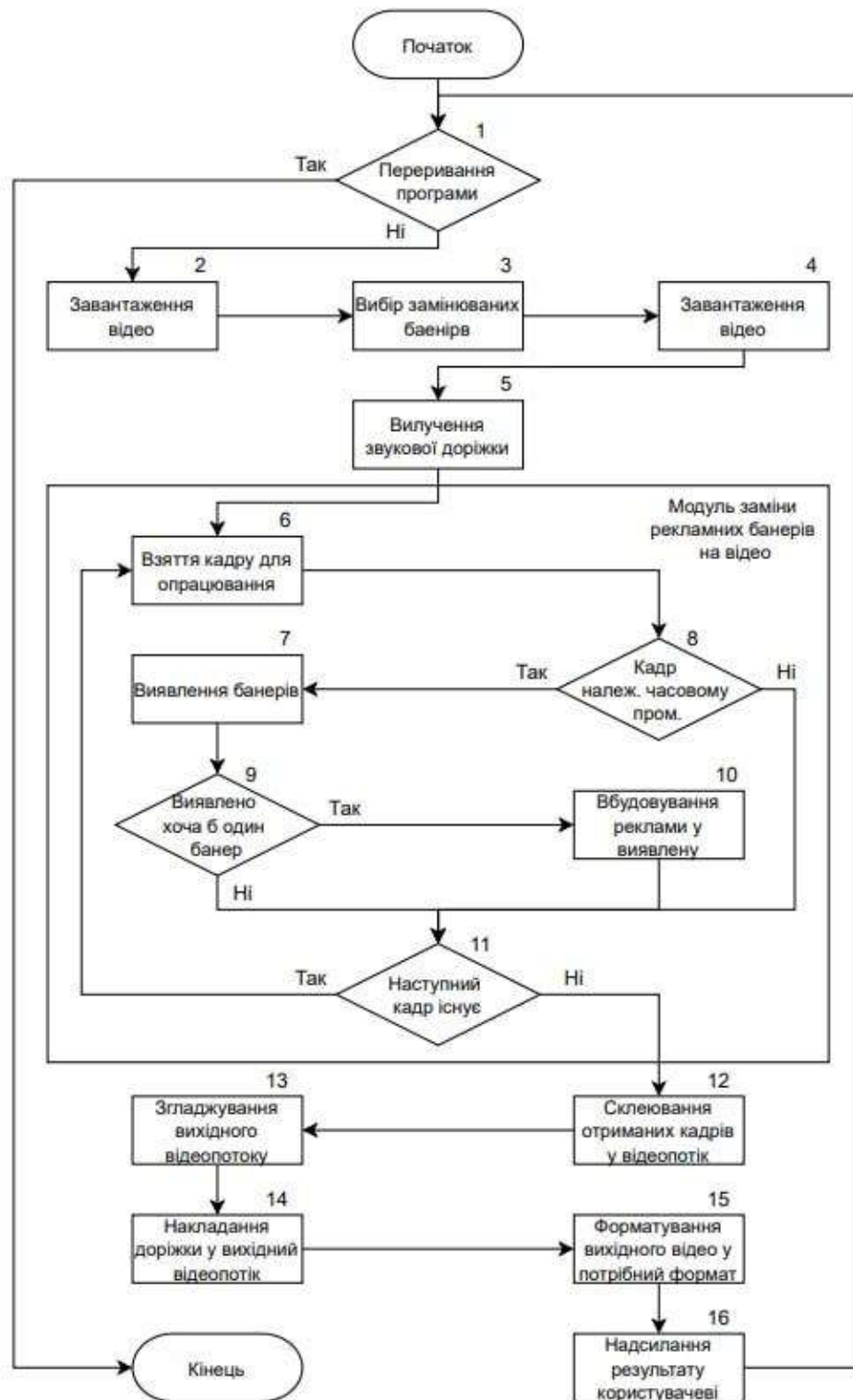


Рисунок В.1 – Алгоритм роботи інформаційної технології заміни рекламних банерів на відео



Рисунок В.2 – Загальна структурна схема інформаційної технології заміни рекламних банерів на відео

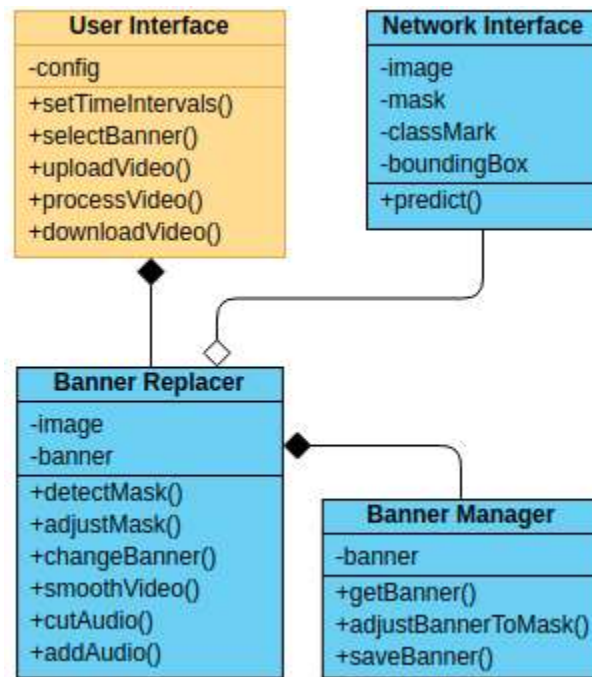


Рисунок В.3 – Діаграма класів інформаційної технології заміни рекламних банерів на відео

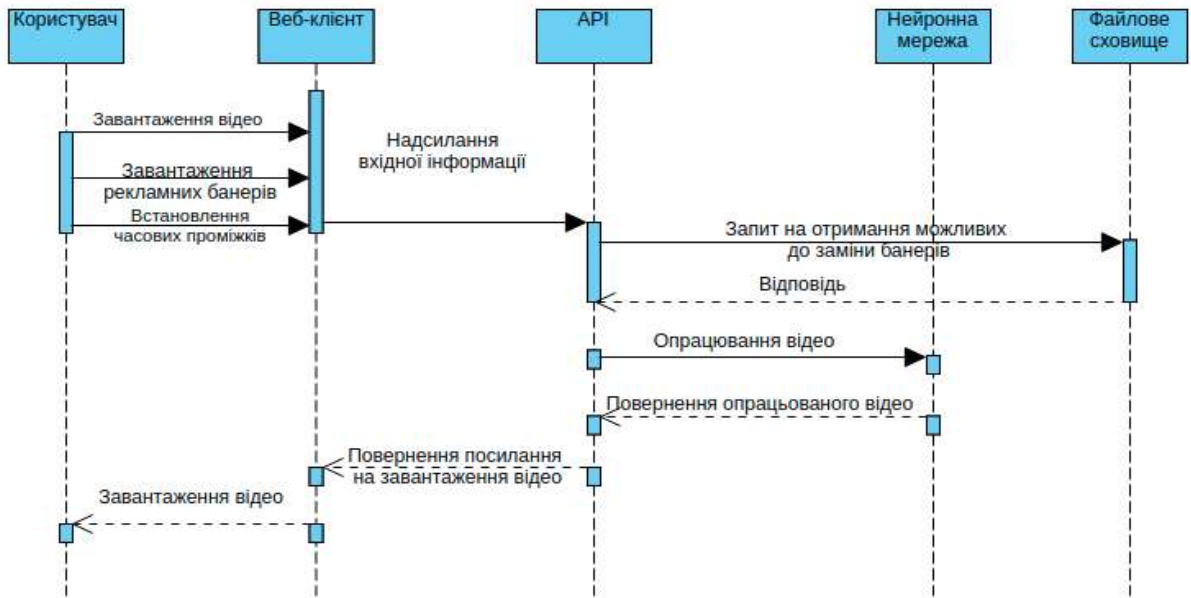


Рисунок В.4 – Діаграма послідовності інформаційної технології заміни рекламних банерів на відео

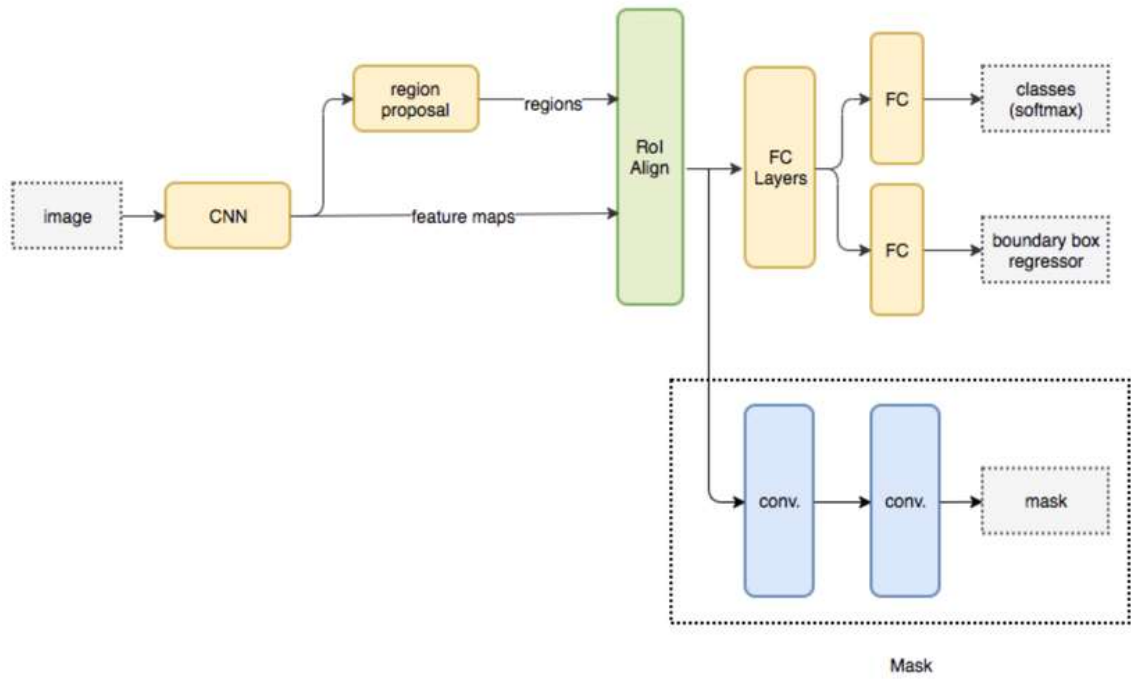


Рисунок В.5 – Архітектура нейронної мережі інформаційної технології заміни рекламних банерів на відео

Period 1

00:00:39

00:00:46

Period 2

00:03:40

00:03:57

Add period **Send**

gazprom

task3.jpg

heineken

No file chosen

mastercard

photo_2021...12-53-25.jpg

nissan

No file chosen

pepsi

No file chosen

playstation

No file chosen

Send

Рисунок В.6 – Робочі вікна інформаційної технології заміни рекламних банерів на відео



Рисунок В.7 – Результат роботи інформаційної технології

Додаток Г (довідниковий)

Інструкція користувача

Було розроблено максимально простий та зрозумілий інтерфейс, щоб користувач міг розпочати працювати з програмою без особливих вмінь та навичок. Програма розташована на локальному комп'ютері та представлена у вигляді веб-додатку, що знаходиться за посиланням <http://0.0.0.0:5089/>. Головна сторінка програмного інтерфейсу передбачає завантаження відео до опрацювання, приклад зображеного на рисунку Г.1.



Рисунок Г.1 – Вигляд головної сторінки веб-клієнта інформаційної технології
заміни рекламних банерів на відео

Після завантаження відео файлу необхідно обрати часові проміжки на яких буде виконуватись заміна рекламних банерів. Потрібно зауважити, у випадку, коли потрібно опрацювати все відео часові проміжки непотрібно встановлювати, як зображено на рисунку Г.2. Виконавши встановлення часових проміжків, користувач може обрати банери, які підлягають заміні та завантажити банери, що потрібно встановити. Приклад вибору замінюваних банерів зображено на рисунку Г.3



Рисунок Г.2 – Вигляд вікна встановлення часових проміжків

gazprom
 4net-resizeimage.jpg

heineken
 1net-resizeimage.jpg

mastercard
 No file chosen

nissan
 photo_2021...10-54-59.jpg

pepsi
 No file chosen

playstation
 django.png

Рисунок Г.3 – Вигляд сторінки вибору замінюваних банерів

Після встановлення параметрів, можна почати опрацювання відео, натиснувши на кнопку «process», що зображено на рисунку Г.4.



Рисунок Г.4 – Вигляд панелі адміністрування після сортування за одним з полів

Після опрацювання відео, сервер зберігає результат в папку results, яка розташована відносно абсолютного шляху й перенаправляє користувача на головну сторінку завантаження відео.

Для завершення роботи із програмою, достатньо закрити вкладку веб-браузер, система автоматично завершить всі процеси.