Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

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

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ СХЕМИ РУХУ З ВИКОРИСТАННЯМ ВІДЕОДЕТЕКТУВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ
(тема)

Виконав:
студент 2 курсу, групи ІНФм-19-1
Дзюбенко М. О.
(прізвище, ініціали)

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

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Творошенко І. С.
(посада, прізвище, ініціали)

Допускається до захисту

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

2020 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)
Рівень вищої освіти другий (магістерський)
Спеціальність 122 Комп'ютерні науки
(код і повна назва)
Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Дзюбенку Микиті Олеговичу
(прізвище, ім'я, по батькові)
1. Тема роботи Дослідження методів аналізу схеми руху з використанням відеодетектування транспортних засобів

затверджена наказом по університету від «_23___» жовтня 2020 року № 1428Ст.
2. Термін подання студентом роботи до екзаменаційної комісії 23 листопада 2020 р.
3. Вихідні дані до роботи Методи аналізу зображень,
перелік використовуваних програмних засобів: C++, OpenCV, Python, Microsoft® Visual Studio 2010, MATLAB R2011,
теоретичні відомості про методи відеодетектування.

4. Перелік питань, що потрібно опрацювати в роботі
1. Огляд методів детектування простих елементів зображення та аналіз їх можливостей.
2. Математична модель відеодетектування транспортних засобів.
3. Структурна схема та опис модулів системи відеодетектування транспортних засобів.
4. Створення системи відеодетектування транспортних засобів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Приклад відображення траєкторії руху транспортного засобу, постановка задачі, тестові зображення, ілюстрації роботи додатку.

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

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

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	23.10.2020	
2	Аналіз завдання, підбір літератури	23.10.20-24.10.20	
3	Аналіз літератури з досліджуваної проблеми	25.10.20-27.10.20	
4	Аналіз технічних засобів	28.10.20-29.10.20	
5	Розробка методу	30.10.20-01.11.20	
6	Програмна реалізація	02.11.20-09.11.20	
7	Оформлення пояснювальної записки	10.11.20-15.12.20	
8	Перевірка на плагіат	23.11.20	
9	Рецензування	24.11.20	
10	Підготовка презентації та доповіді	26.11.20	
11	Занесення роботи в електронний архів	29.11.20	
12	Попередній захист атестаційної роботи	14.12.20	

Дата видачі завдання 23 10 2020 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Творошенко І.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка атестаційної роботи: 70 с., 1 табл., 24 рис., 25 джерел.

ВІДЕОДЕТЕКТУВАННЯ, РОЗПІЗНАВАННЯ, ТРАНСПОРТНІ ЗАСОБИ, НЕЙРОННІ МЕРЕЖІ, OPENCV, PYTHON.

Об’єктом дослідження є схеми руху транспортних засобів, що піддаються класифікації напрямку руху за допомогою засобів відеодетектування.

Мета дослідження полягає в розвитку існуючих методів відеодетектування транспортних засобів для забезпечення стійкості до зміни ракурсу і розташування камери, а також для підвищення якості пошуку і супроводу.

Як середовище розробки було обрано Microsoft Visual Studio, що дозволяє легко компілювати код, а також виводити результати кожного логічного блоку коду.

У результаті роботи здійснено порівняльний аналіз обраних методів для відеодетектування транспортних засобів.

VIDEODETECTION, VEHICLES, RECOGNITION, NEURAL NETWORKS, OPENCV, PYTHON.

The object of the study are traffic diagrams of vehicles that can be classified by direction of movement using video detection.

The purpose of this work is to develop existing methods of video detection of vehicles to ensure resistance to changes in angle and location of the camera, as well as to improve the quality of search and tracking.

Microsoft Visual Studio was chosen as the development environment, which allows you to easily compile the code, as well as display the results of each logical block of code.

As a result, a comparative analysis of selected methods was performed for video detection of vehicles.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 6
Вступ 7
1 Аналіз проблеми відеодетектування транспортних засобів 11
1.1 Математична модель відеодетектування транспортних засобів 11
1.2 Класифікація та аналіз існуючих методів відеодетектування транспортних засобів 15
1.3 Аналіз впливу технологій вирішення задачі відеодетектування транспортних засобів на якість роботи систем відеодетектування об’єктів .. 17
1.4 Постановка задачі дослідження 19
2 Сучасні методи аналізу схеми руху з використанням відеодетектування транспортних засобів 20
2.1 Метод повного покадрового пошуку 20
2.2 Метод покадрового пошуку з відсіканням необхідних областей відповідно до об’єкта дослідження 21
2.3 Метод відеодетектування з врахуванням результатів супроводження на один або декілька кадрів 23
2.4 Метод відеодетектування для відновлення схеми руху 27
2.5 Метод відеодетектування з пошуком на кадрах відеопотоку 30
3 Вибір інструментальних засобів для створення системи відеодетектування транспортних засобів 38
3.1 Вибір інструментальних засобів для створення системи відеодетектування транспортних засобів 38
3.2 Структурна схема та опис модулів системи відеодетектування транспортних засобів 44
3.3 Тестування розробленого програмного засобу 53
3.4 Результати дослідження методів аналізу схеми руху з використанням відеодетектування транспортних засобів 63
Висновок 65
Перелік джерел посилання 67

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

RGB – Red Green Blue

FPS – Frames per second

SVG – Scalable Vector Graphic

ВСТУП

В даний час при вирішенні багатьох практичних завдань використовуються системи комп’ютерного зору. Системи відеоспостереження широко застосовуються для забезпечення безпеки на території різних комплексів і об’єктів (вокзали, аеропорти, торгові та промислові комплекси тощо), зокрема, при вирішенні завдань пошуку залишеного багажу, визначення підозрілих людей по їх траєкторії переміщення. Автоматизовані системи управління процесами використовують методи візуального аналізу зображень або відео в процесі прийняття рішення. Засоби організації графічної інформації застосовують різні підходи для ефективної індексації зображень і відеоданих. Ядром обчислень у всіх перерахованих системах є обробка відео. Оскільки основне застосування результатів комп’ютерного зору зосереджено в області промислової робототехніки, то ключовим стає співвідношення якості і швидкості обробки відеоданих.

Одним з актуальних і практично значущих завдань є проблема якісного і кількісного аналізу транспортних потоків. Статистика продажів автомобілів в країні за даними Асоціації Європейського бізнесу свідчить про щорічне збільшення кількості транспортних одиниць, які виходять на вулиці країни. У зв’язку зі зростанням транспортної навантаження гостро постають питання планування будівництва дорожніх розв’язок та ефективного розподілу транспортних потоків з метою забезпечення безперебійного руху на окремих дорожніх ділянках і магістралях. Вирішення проблеми аналізу потоків забезпечить ефективне автоматичне регулювання руху за допомогою перебудови розкладу перемикання світлофорів на підставі інформації про кількість транспортних одиниць. Наявність статистичної інформації про кількість транспортних засобів різних класів, які перетинають певну дорожню ділянку, дасть можливість оцінювати навантаження на дорожнє полотно з метою прогнозування терміну служби дорожнього покриття та подальшого планування ремонтних робіт. Також дозволить приймати обґрунтовані

рішення про необхідність будівництва доріг-дублерів, проектувати дорожню мережу в масштабах міста. Якісний аналіз дорожніх сцен (положення і клас транспортних засобів, траєкторії переміщення тощо) дозволить частково виявляти порушення правил дорожнього руху: проїзд під заборонні сигнали світлофора, перетин суцільних смуг розмітки, зупинка або стоянка транспорту в заборонених місцях, в'їзд великовантажних автомобілів на шосе з обмеженнями на допустиму вагу транспортної одиниці.

На даний момент комп'ютерний зір є критичною областю в розвитку додатків аналізу транспортних потоків та розробки інтелектуальних транспортних технологій: відеореєстраторів, систем автоматичного паркування, систем автомобільної навігації, систем розпізнавання реєстраційних номерів тощо. Розробка візуальних дорожніх детекторів транспортних засобів до теперішнього часу залишається напрямком, відкритим для наукових і промислових досліджень, оскільки візуальне розпізнавання транспорту забезпечує інтуїтивно зрозумілий спосіб подання інформації на відміну від інших методів.

Відеодетектування – один з можливих підходів до вирішення проблеми аналізу транспортних потоків. Завдання відеодетектування полягає в тому, щоб визначити положення транспортних засобів на кожному кадрі потоку відеоданих і побудувати траєкторію руху виявлених об'єктів. Побудова траєкторії передбачає виділення на наборі послідовно йдучих кадрів сукупності положень, що відповідають кожному об'єкту, який потрапляє в зону захоплення камерою. Складність надійного детектування з використанням методів комп'ютерного зору багато в чому обумовлена тим, що дорожні траси є динамічними «сценами» з різним фоном, що постійно змінюється освітленням і можливим перекриттям об'єктів. В загальному випадку для відеоданих довільної якості зазначене завдання є нерозв'язним, тому що якість багато в чому впливає на обсяг інформації, що отримується.

Основна мета – підвищення якості пошуку за рахунок модифікації алгоритмів комп'ютерного зору. У більшості робіт пошук транспортних

засобів виконується за допомогою виявлення реєстраційних номерних знаків. Цей об’єкт з точки зору алгоритмів розпізнавання є найбільш простим через контрастність фону і символів, обмеженість безлічі символів. Зауважимо, що підхід не дозволяє виявляти транспортні засоби в ситуаціях відсутності номерних знаків (велосипеди) або їх розташування в нестандартних частинах (як у мотоциклів або автомобілів з тимчасовими номерами). При цьому виконання класифікації транспортних засобів потребує доступу до бази даних автоінспекції, що не завжди необхідно і можливо.

Поряд з цим зустрічаються роботи, в яких здійснюється пошук простих геометричних примітивів: прямих ліній в області бампера або підвіски, окружностей, що обмежують колеса, тощо. Відновлення параметрів транспортних засобів здійснюється на підставі взаємного розташування сукупності примітивів та історії їх зміщення в системі координат, пов’язаної з зображенням.

Основні обмеження обох підходів – нестійкість до зміни ракурсу і наявності розмитих кадрів (наприклад, внаслідок короткочасної дії вітру), а також відсутність можливості великоблочної класифікації («АВТОМОБІЛЬ», «АВТОБУС») за одним кадром. У зв’язку з цим існуючі методи не дозволяють вирішувати задачу оцінки інтенсивності транспортних потоків з урахуванням їх якісного складу в умовах, коли транспортні засоби змінюють ракурс, наприклад, при проїзді через перехрестя. Таким чином, проблема розробки методів і алгоритмів рішення задачі відеодетектування транспортних засобів різних класів, що забезпечують пошук і супровід незалежно від ракурсу і місця розташування камери, є актуальною. Це дослідження орієнтоване на обробку відео із стаціонарних камер, на якому транспортні засоби змінюють ракурс і напрям руху; відео, що поступає з відеореєстраторів або отриманого в процесі зйомки дорожнього руху з рук. Рішення можуть використовуватися в системах оцінки інтенсивності транспортного потоку, що перетинає розв’язку в направленні з однієї вулиці до іншої, внаслідок забезпечення більш високої точності; у відеореєстраторах в’їжджаючих та виїжджаючих автомобілів на

автостоянках і паркінгах для контролю завантаження, забезпечуючи нижчу вартість установки при незмінній якості; у модулях класифікації транспортних засобів для ухвалення рішення по одному кадру про приналежність тому або іншому класу, що має значення у разі високих швидкостей.

Мета даного дослідження полягає в розвитку існуючих методів відеодетектування транспортних засобів для забезпечення стійкості до зміни ракурсу і розташування камери, а також для підвищення якості пошуку і супроводу.

Ця мета припускає рішення наступних завдань:

– формулювання напрямів дослідження на основі аналізу існуючих методів відеодетектування транспортних засобів;

– розробка методів відеодетектування транспортних засобів, що реалізовують їх, для відновлення траєкторій руху транспортних засобів, ракурс яких змінюється при русі;

– розробка програмної системи відеодетектування транспортних засобів, яка дозволить виконувати повний цикл рішення задачі відеодетектування, що включає підготовку тестових даних (розмітка положень і траєкторій руху об’єктів на відео, розмітка відсотка видимості об’єктів, розмітка областей інтересу для відео з стаціонарної камери), інтеграцію і застосування різних методів відеодетектування, збір і аналіз отриманих результатів пошуку і супроводу;

– експериментальне дослідження ефективності роботи системи;

– оцінка якості відеодетектування (пошуку і супроводу) для розроблених методів на тестових відео;

Практична значимість і цінність роботи полягає в створенні програмної системи відеодетектування транспортних засобів, що підтримує підготовку тестових даних, інтеграцію і застосування програмних реалізацій розроблених методів, що забезпечують класифікацію за одним кадром, оцінку якості відеодетектування, візуалізацію результатів пошуку і супроводу.

1 АНАЛІЗ ПРОБЛЕМИ ВІДЕОДЕТЕКТУВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ

1.1 Математична модель відеодетектування транспортних засобів

Метод відеодетектування автомобілів працює з потоком відеоданих. Позначимо вхідний відеопотік як *V*. По суті *V* можна представити у вигляді послідовності кадрів або зображень [1]:

$$l_0, l_1, l_2, \dots, l_{N-1},$$

де N – кількість кадрів відеопотоку. Не зменшуючи спільності, вважатимемо, що на вхід поступає кольорове відео у форматі RGB (інакше досить виконати перетворення відео або робочих кадрів в потрібний колірний простір).

RGB – колірна модель, яка описує спосіб формування кольору, як суміші червоного, зеленого і синього кольорів різної інтенсивності. Інтенсивність кожного колірного каналу належить відрізку, внаслідок чого простір RGB представляється у вигляді тривимірного куба з одиничною стороною. Для зручності зберігання в машині інтенсивність позначається цілими числами з відрізання від 0 до 255 (крапка (0, 0, 0) відповідає чорному колір – всі кольори «вимкнені», (255, 255, 255) – білому).

Таким чином, будь-яке зображення послідовності кадрів – двовимірна матриця інтенсивностей пікселів l_k задана у вигляді:

$$l_k = \left(v_{ij}^k\right) 0 \le i \le h-1, 0 \le j \le w-1',$$

де w, h – ширина і висота зображення відповідно, а кожен елемент v_{ij}^k матриці l_k є вектор інтенсивностей по трьом колірним каналах простору:

$$v_{ij}^{k} = \left(r_{ij}^{k}, g_{ij}^{k}, b_{ij}^{k}\right), r_{ij}^{k}, g_{ij}^{k}, b_{ij}^{k} \in \{0,1, \dots ,255\}.$$

Приймемо, що положення об'єкту визначається розташуванням окаймляючого прямокутника. Необхідно відзначити, що залежно від алгоритму під положенням може розумітися розташування контурів об'єкту, особливих крапок (кутів або ребер) або розташування тривимірного геометричного каркаса об'єкту. Завдання відеодетектування транспортних засобів полягає в тому, щоб кожному кадру l_k вихідного відео V поставити у відповідність сукупність положень об'єктів B_k. Тоді завдання відеодетектування зводиться до побудови відображення φ:

$$\varphi: \{l_k, k = 0, \overline{N-1}\} \to \{B_k, k = 0, \overline{N-1}\},$$

де $B_k = \{bb_l^k, l = 0, \overline{s_k - 1}\}$ – множина окаймляючих прямокутників, виявлених на кадрі;

$l_{k,} s_k = |B_k|$ – кількість знайдених прямокутників.

При цьому кожен прямокутник bb_l^k визначається набором наступних компонентів:

$$bb_l^k = \left(\left(x_1^l, y_1^l\right), \left(x_2^l, y_2^l\right), [s^l, c^l]\right),$$

де $\left(x_1^l, y_1^l\right), \left(x_2^l, y_2^l\right)$ – координати лівого верхнього і правого ніжнего кутів прямокутника $\left(0 \le x_1^l \le x_2^l \le w-1, 0 \le y_1^l \le y_2^l \le h-1\right), s^l \epsilon \mathbb{R}$ – достовірність того, що об'єкт виявлений правильно, а $c^l \epsilon\, C = \{'CAR', 'BUS', \dots, 'TRAIN'\}$ – клас, якому належить транспортний засіб (рис. 1.1). Відмітимо, що залежно від специфіки вирішуваного завдання і вибору методу її рішення останні два компоненти можуть бути відсутніми.

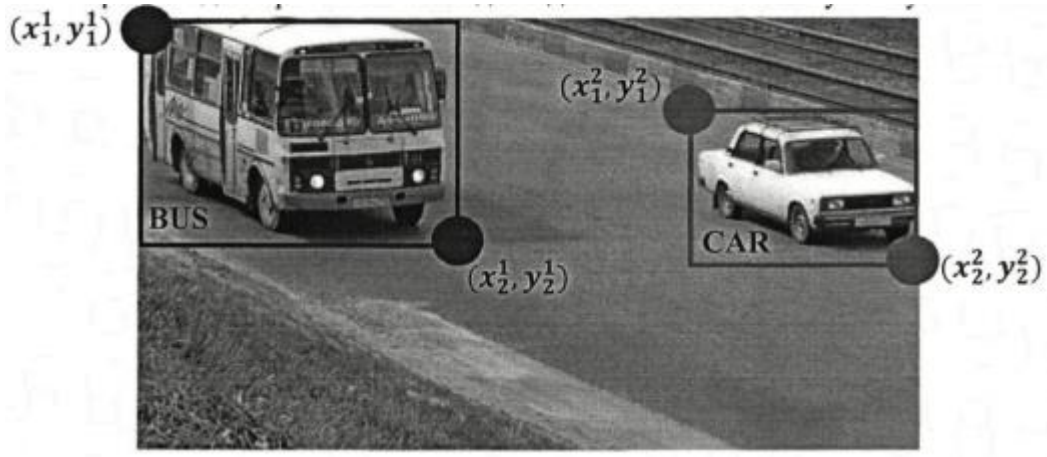

Рисунок 1.1 – Набір окаймляючих прямокутників

Складність завдання пошуку об’єктів на зображенні обумовлена багатьма причинами, серед яких можна виділити декілька основних:

– різноманітність форм і кольорів представників класу об’єктів. В разі безлічі транспортних засобів це різноманітність типів (легкові і вантажні автомобілі, автобуси, мотоцикли і так далі), моделей і кольорів;

– перекриття об’єктів, що детектуються. Природна ситуація, коли з точки зору камери об’єкти видно частково, наприклад, оглядається лише дах і вітрове скло автомобіля;

– різна міра освітленості об’єктів, обумовлена часом і і погодними змінами.

Результат роботи вибраного алгоритму не завжди залежить лише від інформації, витягуваної з поточного кадру. У загальному випадку відображення залежить від набору параметрів, серед яких:

– набір окаймляючих прямокутників для об’єктів, виявлених на $l_1 -$ попередніх кадрах $l_{k-1}, l_{k-l+1}, \dots, l_{k-1}$ відео $V(l_2 \leq N - 1 - k)$;

– набір окаймляючих прямокутників для об’єктів, виявлених на а $l_2 -$ наступних кадрів $l_{k+1}, \dots, l_{k+l2}$ відео $V(l_2 \leq N - 1 - k)$;

– сукупність параметрів $p^{(d)}$ алгоритму пошуку об’єктів;

– набір параметрів $p^{(t)}$ алгоритму супроводу.

Якщо розглядається завдання відеодетектування, до якого внесені спеціальні вимоги, то поряд з вказаним відображенням φ, як правило, потрібно побудувати додаткові відображення. У даній постановці необхідно виконувати побудову траєкторій руху об'єктів і підрахунок транспортних засобів з врахуванням напряму руху.

Введемо відображення ψ для зіставлення окаймляючих прямокутників парі послідовних кадрів l_k і l_{k+1}. Звернемо увагу, що на деякому кадрі об'єкт зникає з поля зору камери. Звернемо увагу, що на деякому кадрі об'єкт зникає з поля зору камери. Тому слід розглядати безліч образів, розширену спеціально виділеним окаймляючим прямокутником,

$$bb = ((-1,-1),(-1,-1),[s,c]),$$

де s – довільна речова величина;

c – довільний клас об'єктів.

Відзначимо, що в даному випадку може бути вибраний будь-який інший прямокутник з неіснуючими координатами. Тоді ψ представляє відображення вигляду

$$\psi: B_k \to B_{k+1} \cup \{bb\}, \forall k = 0, \overline{N-2}.$$

Допустимо, що l_k – перший кадр, на якому був виявлений об'єкт, що міститься в прямокутнику з індексом $r_0(k) \in \{0,1,2,\dots,s_k\}$, – кількість подальших кадрів, на яких об'єкт видно спостерігачеві. Тоді траєкторією (треком) руху $T^k_{r_0(k)}$ транспортного засобу називається послідовність положень відповідних окаймляючих прямокутників.

$$T^k_{r_0(k)} = (bb^k_{r_0,} bb^{k+1}_{r_1}, bb^{k+2}_{r_2}, \dots, bb^{k+q-1}_{r_{q-1},}),$$

де $\in \{0,1,2,\dots,s_{k+i}\}, bb_{r_i}^{k+i} = \psi(bb_{r_{i=1}}^{k+i-1}), i = 1,\overline{q-1}.$

Далі пропонується огляд найбільш поширених методів комп'ютерного зору для вирішення завдання відеодетектування, яке включає пошук, супровід і класифікацію транспортних засобів.

1.2 Класифікація та аналіз існуючих методів відеодетектування транспортних засобів

Множина всіх методів детектування транспортних засобів, що пересікають дорожню ділянку, можна розділити на дві групи: «інвазивні» і «неінвазивні» [2]. «Інвазивні» методи передбачають втручання в дорожнє покриття за допомогою вбудовування технічних приладів. Серед найбільш поширених апаратних рішень можна виділити:

– індуктивні детектори – індуктивні ланцюги, які прокладаються під дорожнім покриттям, ланцюги реагують на появу металевих об'єктів на даній дорожній ділянці;

– пневматичні дорожні канали – пневматичні стрічки, що прокладаються впоперек дорожніх смуг поверх покриття, реагують на вагу автомобілів;

– пневматичні дорожні канали – пневматичні стрічки, що прокладаються впоперек дорожніх смуг поверх покриття, реагують на вагу автомобілів;

– п'єзоелектричні сенсори – сенсори, в основі яких лежить явище прямого п'єзоефекту (механічна дія, прикладена певним чином до п'єзоелектричного кристала, породжує електричну напругу).

«Неінвазивні» методи дозволяють використовувати апаратуру, встановлену в межах видимості дорожньої ділянки, а також програмне

забезпечення для обробки інформації, що поступає. До «неінвазивних» рішень можна віднести:

– інфрачервоні сенсори – сенсори, що визначають присутність об’єкту, його швидкість і типа транспортного засобу на підставі випромінюваної енергії в зоні детектування. Основним недоліком такого роду рішень є непристосованість до суворих кліматичних умов;

– мікрохвильові радари – радари, які дозволяють визначати швидкість рухомих об’єктів на підставі здатності електромагнітних хвиль проникати в оптично непрозорі середовища і частково відбиватися від будь-яких неоднородностей на дорозі поширення. Мікрохвильові радари не дозволяють знаходити статичні або близько розташовані один до одного об’єкти;

– ультразвукові датчики – пристрої, в яких для визначення кількості транспортних засобів і їх типів використовуються звукові хвилі і звукова енергія. На ультразвукові датчики роблять значний вплив температурні режими і явища турбулентності.

Перераховані рішення (як інвазивні, так і неінвазивні) використовують спеціалізоване апаратне забезпечення, що має ряд обмежень, – від високої вартості, складності супроводу і впровадження до недовгого терміну служби. Далі в роботі розглядаються системи відеодетектування, найбільш перспективні з точки зору об’єму інформації, що надається, і як наслідок, і практичного використання.

Системи відеодетектування реалізуються програмним чином на сервері обробки потоку відеоданих або інтегруються безпосередньо в камеру, оснащену програмованим сигнальним процесором або мікросхемою (applicationspecific integrated circuit) [3]. Процесори дозволяють визначити присутність об’єкту (у тому числі і нерухомого) в межах смуги руху, обчислити його розмір, проклассифіцировать об’єкт на підставі співвідношення його довжини і ширини. У вказаних системах також може бути використана додаткова інформація, що поступає з різних датчиків (наприклад, дані з GPS дозволяють визначити положення камери на карті

міста). Сучасні системи відеодетектування поставляються в прив’язці до устаткування, а розробники програмних модулів ведуть адаптацію наявних бібліотек для інших камер і відеопроцесорів. При цьому в бібліотеках, як правило, реалізуються методи пошуку простих об’єктів, які є частямі транспортних засобів, наприклад, реєстраційних номерів або коліс. Як наслідок, накладаються обмеження на дозвіл відеокамер і їх розташування в межах дороги. Природне бажання розробників систем відеодетектування полягає в тому, щоб підвищити відмовостійкість і якість роботи системи при зміні розташування відеокамери і конфігурації апаратного забезпечення.

Один з пріоритетних напрямів досліджень – створення нових і адаптація існуючих методів відеодетектування на відеоданих невисокого розширення.

1.3 Аналіз впливу технологій вирішення задачі відеодетектування транспортних засобів на якість роботи систем відеодетектування об’єктів

Можна виділити два основні підходи до побудови повнофункціональних систем відеодетектування транспортних засобів: пошук і супровід областей руху [4] з подальшою ідентифікацією цих областей як транспортних засобів того або іншого класу («АВТОБУС», «АВТОМОБІЛЬ», «МОТОЦИКЛ» і тому подібне); пошук і супровід виявлених транспортних одиниць. При реалізації пошуку і супроводу областей руху роботу системи відеодетектування можна розбити на декілька етапів:

1) Виділення областей руху на поточному кадрі відео.

2) Супровід областей руху на декількох подальших кадрах. В процесі супроводу забезпечується побудова траєкторії руху для кожної області. Відзначимо, що в результаті перекриттів кожна область може містити групи об’єктів.

3) Відновлення параметрів об’єктів: розділення об’єктів, що належать одній області руху, класифікація транспортних засобів, визначення довжини і

ширина об’єкту і інших параметрів, яку можна обчислити на підставі отриманої послідовності областей руху і інформації про положення камери.

4) Супровід транспортних засобів. На даному етапі виконується побудова траєкторій руху транспортних засобів, виділених на попередньому кроці, від моменту їх ідентифікації до їх виходу з області огляду камери.

В даний час більшість відомих систем відеодетектування функціонують згідно з даною схемою. Відмітимо, що схема, заснована на пошуку і супроводі областей руху, працює виключно в разі нерухомої камери і постійного фону. У природних умовах камера завжди знаходиться під впливом вітру і опадів, тому нерухомість практично недосяжна. У таких ситуаціях зазвичай застосовується програмна стабілізація відео.

Схема роботи систем відеодетектування за допомогою виявлення і супроводу об’єктів не залежить від типа об’єктів (людина, автомобіль, трамвай і ін.), що детектуються, і складається з наступної послідовності дій:

1. Витягання кадру з потоку відеоданих.
2. Передобробка отриманого статичного зображення – кадру.
3. Пошук положення об’єкту на зображенні і оцінка міри достовірностізнаходження об’єкту в даної області. На додаток на даному етапі в деяких системах виконується внутрікласове ділення – класифікація об’єктів в рамках одного даного класу.
4. Супровід виявлених об’єктів.
5. Аналіз результатів пошуку і супроводу (наприклад, оцінка напряму руху об’єкту на підставі траєкторії, підрахунок числа об’єктів і тому подібне).

Функціональність блоку аналізу результатів в основному залежить від конкретного завдання, яке поставлене перед розробниками системи відеодетектування. При реалізації приведеної схеми ядром системи є модулі пошуку і супроводу об’єктів. Схема є обчислювально більш трудомісткою в порівнянні з попередньою, але в процесі її вживання не накладаються обмеження на відеодані, тобто можливе використання даних, що поступають з камер, встановлених на автомобілях, оскільки приведений підхід забезпечує

незалежність процедури пошуку від структури фону. Далі зупинимося на деяких найбільш поширених методах пошуку і супроводу.

1.4 Постановка задачі дослідження

Об’єктом дослідження є схеми руху транспортних засобів, що піддаються класифікації напрямку руху за допомогою засобів відеодетектування.

Мета дослідження полягає в розвитку існуючих методів відеодетектування транспортних засобів для забезпечення стійкості до зміни ракурсу і розташування камери, а також для підвищення якості пошуку і супроводу.

Ця мета припускає рішення наступних завдань:

– формулювання напрямів дослідження на основі аналізу існуючих методів відеодетектування транспортних засобів;

– розробка методів відеодетектування транспортних засобів, що реалізовують їх, для відновлення траєкторій руху транспортних засобів, ракурс яких змінюється при русі;

– розробка програмної системи відеодетектування транспортних засобів, яка дозволить виконувати повний цикл рішення задачі відеодетектування, що включає підготовку тестових даних (розмітка положень і траєкторій руху об’єктів на відео, розмітка відсотка видимості об’єктів, розмітка областей інтересу для відео з стаціонарної камери), інтеграцію і застосування різних методів відеодетектування, збір і аналіз отриманих результатів пошуку і супроводу;

– експериментальне дослідження ефективності роботи системи;

– оцінка якості відеодетектування (пошуку і супроводу) для розроблених методів на тестових відео;

2 СУЧАСНІ МЕТОДИ АНАЛІЗУ СХЕМИ РУХУ З ВИКОРИСТАННЯМ ВІДЕОДЕТЕКТУВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ

2.1 Метод повного покадрового пошуку

У простому випадку відео розглядається як послідовність незв'язаних зображень. Тому пошук об'єктів можна проводити незалежно на кожному кадрі відео. Результатом пошуку є набір окаймляючих прямокутників [5]

$$B_k = \left\{ bb_l^k, l = 0, \overline{s_k - 1} \right\}, \forall k = \overline{0, \mathrm{N} - 1}.$$

У роботі пропонується використовувати пошук об'єктів різних класів Latent SVM, заснований на визначенні положень зв'язаних частин об'єкту і побудові моделі класу за допомогою методів машинного вчення. Даний алгоритм дозволяє знаходити транспортні засоби наступних класів: «АВТОМОБІЛЬ», «АВТОБУС», «МОТОЦИКЛ», «ВЕЛОСИПЕД», «ТРАМВАЙ». Приведений перелік класів визначається набором моделей, отриманих на етапі вчення алгоритму, і потенційно може бути розширений за рахунок вчення нових моделей. Вчення нових моделей вимагає підготовки розміченої бази зображень. Можлива перспективність вживання алгоритму Latent SVM до рішення нової задачі відеодетектування підтверджується результатами конкурсу VOC 2007. Програмна реалізація алгоритму пошуку була виконана раніше автором на початку наукового дослідження. Для підвищення ефективності роботи реалізації розроблена схема розпаралелювання, яка дозволила отримати зниження середнього часу обробки одного зображення. В середньому прискорення складає 3,3 разу на чотирьох потоках, час обробки зображення з дозволом 640×480 було зменшено в середньому до 1,28 с.

Вказане прискорення – непоганий результат за умови, що распараллелени не всі етапи алгоритму пошуку, а лише найбільш трудомісткий, який займає орієнтування 80% часу від загального часу пошуку об’єктів заданого класу на зображенні. Можливий недолік схеми повного покадрового детектування – наявність великої кількості помилкових спрацьовувань з точки зору вживання в практичних застосуваннях, тобто положень, які реально не містять об’єкти, що цікавлять. Зв’язано це з тим, що класифікатор може бути виучений на даних, які не корелюють з тестовими зображеннями об’єктів на відео. Необхідно відзначити, що даний метод не дозволяє будувати траєкторії руху транспортних засобів, а як слідство, не дає повного рішення задачі відеодетектування.

2.2 Метод покадрового пошуку з відсіканням необхідних областей відповідно до об’єкта дослідження

Один з широко відомих підходів до зниження кількості помилкових спрацьовувань детектора [6], вживаних при розробці систем відеоспостереження, полягає в тому, щоб відсікти положення, що знаходяться поза областями інтересу. При рішенні задачі відеодетектування транспортних засобів необхідно вказати на відео області дороги, в яких потенційно можуть знаходитися об’єкти в процесі зйомки, – області інтересу. Область інтересу задається бінарним зображенням:

$$R = (r_{ij})_{0 \le i \le h-1, 0 \le j \le w-1},$$

де $r_{ij} \in \{0,255\}$;

$r_{ij} = 255$ – білий піксель;

$r_{ij} = 0$ – чорний піксель.

Зображення R може бути отримане уручну один раз для заданого відео або за допомогою сегментації (рис. 2.1). Якщо перший спосіб застосовний за умови, що камера встановлена статично в області видимості дороги, то другою може бути використаний і в разі рухливої камери.

Рисунок 2.1 – Приклад областей інтересу

Алгоритм відеодетектування з відсіканням по областях інтересу для кадру з номером до складається з двох етапів:

– детектування транспортних засобів – побудова множини положень

$$B_k = \{bb_l^k, l = 0, \overline{s_k - 1}\};$$

– відсікання по області R. Приймається, що виявлений прямокутник належить області інтересу, якщо виконується наступне співвідношення:

$$\frac{p_w}{p_b} \geq t,$$

де p_w – кількість пікселів, що належать області інтересу і накриваних виявленим прямокутником;

p_b – число пікселів, лежачих усередині прямокутника, але що не належать області інтересу;

t – порогове значення.

Відмітимо, що вибір параметра *t* залежить від точності розмітки областей інтересу. Якщо в процесі розмітки враховувалося, що об'єкт частково може виходити з області дороги (наприклад, коли автомобіль має значну висоту, частина кузова на зображенні може виходити за межі області інтересу), то має сенс вибирати велике порогове значення. Відповідно, чим точніше до кордонів дорожнього полотна були виділені області інтересу, тим менше вказаний поріг відсікання. Такий підхід дозволяє значно понизити кількість помилкових положень, розташованих зовні областей інтересу, але не позбавить від помилкових спрацьовуванні алгоритму пошуку усередині цих областей.

2.3 Метод відеодетектування з врахуванням результатів супроводження на один або декілька кадрів

До справжнього моменту кадри розглядалися незалежно, і завдання детектування вирішувалося окремо на кожному зображенні. Природно, що послідовні кадри відео зв'язані між собою і транспортні засоби не можуть з'являтися в довільних частинах зображення. Ідея вживання трекінгу на один кадр вперед полягає в тому, щоб забезпечити додатковий рівень перевірки коректності роботи детектора з метою відсікання помилкових спрацьовувань.

На даному етапі дослідження пропонується новий метод відеодетектування транспортних засобів [7], що враховує результати однокрокового трекінгу. Розглянемо алгоритм роботи методу. Алгоритм пошуку запускається на парі кадрів, що послідовно йдуть. Позначимо індекси відповідних кадрів *i* та *i + 1*.

Для опису схеми введемо наступні позначення:

– B_i та B_{i+1} – множина продетектованих прямокутників на кадрах *i* та *i+1*;

– T^{i+1} – множина відновлених положень прямокутників в результаті B_i;

– K^i – множина ключових крапок, побудованих усередині окаймляючих прямокутників набору B_i;

У припущенні, що множина B_i побудовано, обчислювальну схему можна представити у вигляді послідовності наступних етапів:

1) Виявити транспортні засоби на кадрі $i+1$ – побудувати множина B_{i+1} (рис. 2.2).

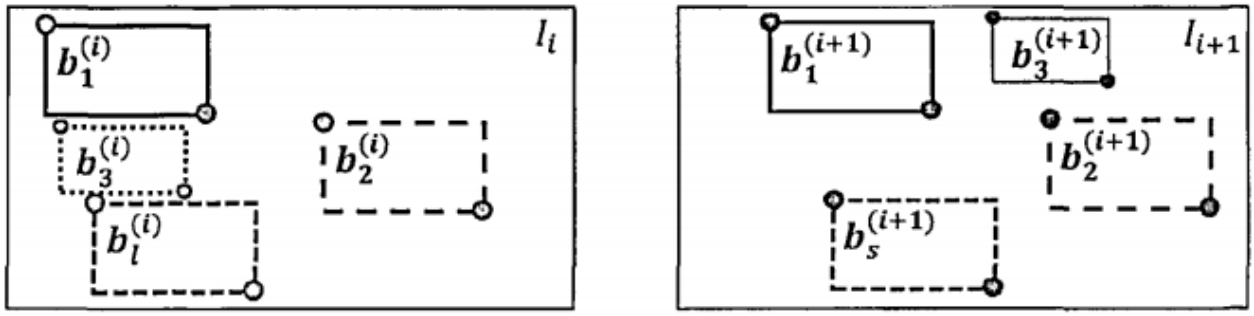

Рисунок 2.2 – Пошук транспортних засобів на кадрі l_{i+1}

2) Відновити положення прямокутників B_i на кадрі I_{i+1}, застосувавши алгоритм супроводу до побудованого набору ключових точок K^i . По суті алгоритм трекінгу дозволить відновити множину K^{i+1}. Потім досить обчислити нові положення прямокутників T^{i+1}. Нові положення можна отримати, виходячи із зсуву довільної ключової крапки, яка має образ в K^{i+1} (рис. 2.3).

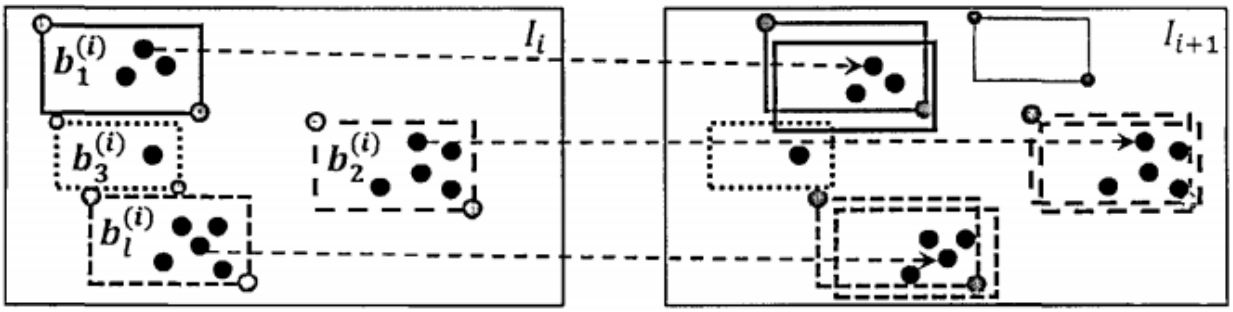

Рисунок 2.3 – Відновлення положень прямокутників B_i на кадрі I_{i+1}

3) Поєднати результати детектування B_{i+1} і супроводи T^{i+1}. Для цього для кожного класу продетектованих об’єктів необхідно упорядкувати прямокутники з множини B_{i+1} по убуванню міри достовірності того, що об’єкт був виявлений правильно. Потім перебрати всі відновлені прямокутники для заданого класу об’єктів з і визначити той, який має максимальний відсоток перекриття з вибраним продетектованим прямокутником. Якщо максимальний відсоток більше деякого заданого порогу (параметр методу), то вважаємо, що це положення одного і того ж об’єкту. Якщо для якого-небудь прямокутника з множини T^{i+1} не виявилася відповідність з B_{i+1} то приймається, що відповідний прямокутник з B_i був виявлений помилково.

Відзначимо, що дана схема також може включати відсікання по областях інтересу на етапі пошуку об’єктів. Результати експериментів показують зниження числа помилкових спрацьовувань при незначному падінні якості пошуку. Даний факт пояснюється тим, що алгоритм пошуку на кадрі *i+1* міг пропустити об’єкт, як наслідок, і на кадрі *i* наявність об’єкту також не буде підтверджена. Подібна ситуація може виникати, наприклад, при частковому виході об’єкту з області огляду камери або перекритті іншими об’єктами, тобто у випадках недостатньої видимості об’єкту. Основна умова застосовності даного методу – об’єкт повинен мати прийнятні розміри, щоб можна було виділити характерні ознаки для подальшого вживання якого-небудь алгоритму супроводу. Як наслідок, такий метод може бути використаний для аналізу відео, що містить транспортні засоби , геометричні параметри яких перевищують сповна певні значення. Також необхідно відзначити, що вживання якісних алгоритмів супроводу (наприклад, пірамідальна модифікація алгоритму Лукасу-Канаді) вимагає ретельного підбору параметрів.

Метод відеодетектування із застосуванням трекінгу полягає в тому, щоб виконувати детектування не на парі послідовно кадрів, що йдуть, а на кадрах, розташованих з деяким кроком у відеопотоці. При цьому положення окаймаючих прямокутників на проміжних кадрах можна відновлювати на

підставі результатів супроводу з використанням існуючих алгоритмів трекінгу. Тоді значення ступені достовірності для прямокутників на проміжних кадрах слід довизначати значеннями, отриманими в результаті детектування на поточному кадрі. Таку процедуру умовно можна розглядати як інтерполяцію.

Позначимо номери кадрів, на яких буде виконуватися детектування, як і та $i + step$, де $step$ – крок, параметр алгоритму. Далі припустимо, що положення транспортних засобів на кадр I_t побудовані, тобто є множина B_t.

Тоді обчислювальну схему можна представити у вигляді наступної послідовності дій:

1) Виявити транспортні засоби на кадрі I_{i+step} – побудувати множину прямокутників B_{i+step} (рис. 2.4).

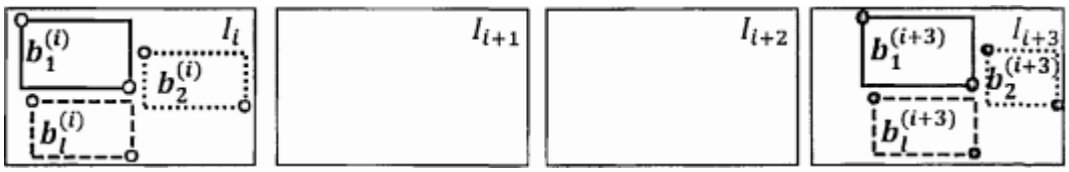

Рисунок 2.4 – Приклад виконання пошуку об’єктів для випадку $step = 3$

2) Побудувати множину ключових точок K^i для заданого набору продетектованих прямокутників B_i.

3) Відновити положення прямокутників B_i на кадрах $i + 1$, $i + 2$,.... $i + +step$, застосувавши алгоритм супроводу до збудованого набору особливих точок K^i. По суті алгоритм трекінгу дозволить відновити множини $K^{i+1}, K^{i+2}, \dots, K^{i+step}$, потім досить обчислити нові положення прямокутників – побудувати $T^{i+1}, T^{i+2}, \dots, T^{i+step}$ (рис. 2.5). Положення можна отримати аналогічно схемі з кроком в один кадр.

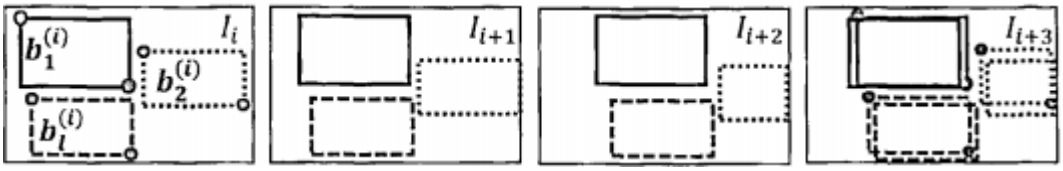

Рисунок 2.5 – Приклад результатів відновлення прямокутників для *step=3*

4) Поєднати результати детектування і супроводу на кадрі i + *step* – зіставити множини B^{i+step} и T^{i+step} . Алгоритм зіставлення такий же, як і в разі однокрокової схеми.

Основна проблема останніх двох схем полягає в тому, що найменш трудомісткі алгоритми супроводу засновані на обчисленні оптичного потоку, велика частина яких дає неякісний результат у разі високих швидкостей руху об’єктів і потребує ретельного підбору параметрів алгоритму супроводу залежно від значення кадрової частоти (загальноприйнята одиниця виміру – FPS).

Зауважимо, що вибір кроку чергового детектування багато в чому залежить від зазначеної характеристики та середніх швидкостей, допустимих на дорожній ділянці, де встановлена камера. Експерименти показали, що при використанні пірамідального алгоритму Лукаса-Канаді з параметрами за замовчуванням і частоті відео, що дорівнює 25 кадрам в секунду, найкращий крок становить три кадри. Збільшення кроку призводить до значного падіння якості пошуку.

2.4 Метод відеодетектування для відновлення схеми руху

Всі описані раніше обчислювальні схеми за винятком, останньої дозволяють побудувати множину положень об’єктів на кожному кадрі відео, але не траєкторії руху. При наявності набору положень, виявлених на всіх кадрах відео, траєкторії руху об’єктів слід будувати окремим чином. Поряд з побудовою траєкторій можна виконувати коригування множини виявлених

прямокутників, а саме відкидання помилкових спрацьовувань детектора, які повторюються на коротких послідовностях кадрів. Далі описується метод відновлення траєкторій з корекцією результатів пошуку об’єктів, розроблений в рамках наукового дослідження. Метод працює в режимі постобробки побудованого набору положень [8] і дозволяє отримати повне рішення завдання відеодетектування.

Допустимо, що побудована множина положень $B_0, B_1, \dots, B_{N-1}$ на всіх кадрах відео за допомогою використання одного з методів. Тоді алгоритм відновлення траєкторій і корекції виявлених прямокутників включає чотири основні етапи:

1) Розбиття сукупності всіх прямокутників на невеликі траєкторії. Приймається, що на послідовності зображень, що містять траєкторію, об’єкт не міняє різко напрям руху.

2) Відновлення «пропусків» детектора усередині побудованих траєкторій.

3) Злиття невеликих траєкторій – об’єднання траєкторій одного і того ж об’єкту, що міняє напрям руху.

4) Відкидання коротких траєкторій. Розглянемо детальніше алгоритм роботи кожного приведеного етапу.

5) Спочатку об’єднаємо множини $B_0, B_1, \dots, B_{N-1}$ у єдину множину B, тоді кожен елемент $b_i \in B$ визначається набором з декількох компонент:

$$b_i = \left(f_i, \left(x_1^i, y_1^i\right), \left(x_2^i, y_2^i\right), s^i c^i\right), \forall i = \overline{0, |B_0 \cup B_1 \cup \dots \cup B_{N-1}| - 1},$$

де f_i – індекс кадру, на якому виявлений об’єкт;

$\left(x_1^i, y_1^i\right), \left(x_2^i, y_2^i\right)$ – координати лівого верхнього і правого нижнього кутів окаймляючого прямокутника;

s^i – достовірність присутності об’єкту в даному прямокутнику;

c^i – клас транспортного засобу.

Таким чином, на вході першого етапу алгоритму є множина B. На даному кроці завдання полягає в тому, щоб сформувати сукупність невеликих траєкторій T^1:

$$\forall t_k \in T^1, t_k = \left(b_{r_0}, b_{r_1}, \dots, b_{r_{s_{k-1}}}\right), \forall k = \overline{0, |T^1| - 1},$$

де $b_{r_j} \in B, \forall j = \overline{0, k}, f_{r_0} < f_{r_1} < \cdots < f_{r_{s_k}}$, s_k – довжина траєкторії.

Відмітимо, що пара положень, що послідовно йдуть, в траєкторії не обов'язково знаходяться на послідовних кадрах відео, оскільки детектор здатний пропускати об'єкти.

Отже, етап розбиття на траєкторії включає декілька кроків:

1) Витягувати положення b_0 з множини B: $B' = B \backslash \{b_0\}$.

2) Вважати положення b_0 початком нової траєкторії t=(b_0).

3) Проглянути все подальші положення $b'_i, \forall i = \overline{0, |B'| - 1}$ і знайти на найближчому наступному кадрі прямокутник, відповідний b_0.

4) Виконати побудову частини трека, що залишився. Якщо в деякий момент побудована частина трека $t = (b_0, b'_{q1}, b'_{q2}, \dots, b'_{qk})$, тоді, щоб відновити наступне положення об'єкту, слід реалізувати послідовність дій, подібну тій, що була описана на попередньому кроці. Різниця лише в тому, що при виборі положення з максимальною кількістю відповідностей для кадру з фіксованим номером додатково слід контролювати напрям зсуву об'єкту v^{curr}. Якщо кут між напрямами v^{prev} та v^{curr} більше порового значення τ^{α}, тоді це означає різку зміну напряму руху об'єктом. При розбитті на траєкторії вважається, що це інший об'єкт, і положення не розглядається при виборі найкращої відповідності.

5) Додати отриману траєкторію t в результуючу множину T^1.

6) Якщо множина B', тоді необхідно повторити всі кроки, починаючи з кроку.

7) Очевидно, що при використанні такої схеми розбиття в траєкторіях, по-перше, можна буде виявити «пропуски», що обумовлено тим, що детектор на деяких кадрах не знаходить об’єкт. По-друге, автомобілі в процесі руху здатні різко змінювати напрямок руху, наприклад, внаслідок перебудування в іншу смугу руху, або при повороті у вулицю з іншим напрямком руху, або при включенні задньої передачі. Як наслідок, траєкторія руху об’єкта може бути розірвана на кілька невеликих траєкторій, що містяться в множині T^1.

Проблема наявності «пропусків» частково вирішується на другому етапі методу – етапі відновлення положень об’єкту [9]. На вході є множина траєкторій T^1, побудоване на попередньому етапі. В результаті відновлення необхідно отримати оновлену множину T^2, причому кількість траєкторій не міняється, тобто $|T^1| = |T^2|$.

2.5 Метод відеодетектування з пошуком на кадрах відеопотоку

На даному етапі дослідження розробляється метод відеодетектування, що забезпечує пошук транспортних засобів на деяких кадрах відео і одночасну побудову траєкторій руху. Мета розробки цього методу – зменшити кількість кадрів, на яких пошук транспортних засобів виконується як на незалежних зображеннях, при цьому одночасно виконувати побудову траєкторій руху з використанням методів зіставлення плоских зображень текстурних об’єктів. Пропонований метод відеодетектування використовує деякі підходи, запропоновані в схеми детектування з урахуванням результатів супроводу на кілька наступних кадрів і алгоритму відновлення траєкторій з корекцією результатів відеодетектування. Метод базується на ідеї двопрохідності відео (у прямому і зворотному часі), запропонованої в алгоритмі медіанного потоку (Median Flow, Predator) [10] для супроводу унікальних об’єктів.

Перейдемо до опису розробленого методу. На кожній ітерації розглядається блок з *step* послідовно ідуть зображень. При цьому алгоритм детектування запускається на першому і останньому кадрі блоку, далі виконується зіставлення результатів детектування і відновлення положень транспортних засобів на проміжних кадрах.

Для опису методу введемо ряд допоміжних позначень: b_i – набір положень об’єктів, продетектованих на кадрі з номером *i*:

$$\forall b_k^{(i)} \in B_i, b_k^{(i)} = \left(f_k^{(i)}, x_k^{(i)}, y_k^{(i)}, w_k^{(i)}, h_k^{(i)}, c_k^{(i)}, s_k^{(i)}\right), k = \overline{0, |B_l| - 1},$$

де $f_k^{(i)}$– індекс кадру;

$(x_k^{(i)}, y_k^{(i)})$ – координати лівого верхнього кута прямокутника;

$(w_k^{(i)}, h_k^{(i)})$ – ширина і висота окаймляючого прямокутника;

$c_k^{(i)}$ – клас транспортного засобу;

$s_k^{(i)}$ – достовірність знаходження об’єкту у відповідному прямокутнику.

$$B_j' \forall b_l^{(i)\prime} \in B_j', b_l^{(i)\prime} = \left(f_l^{(i)\prime} x_l^{(i)\prime} y_l^{(i)\prime} w_l^{(i)\prime} h_l^{(i)\prime} c_l^{(i)\prime} s_l^{(i)\prime} t_l^{(i)\prime}\right), l = \overline{0, \left|B_j'\right| - 1},$$

де $t_l^{(i)\prime}$ – ідентифікатор траєкторії або номер об’єкту на відео, а останні позначення збігаються з позначеннями в $B_j \cap B_j' = \emptyset$;

B_j' – множина положень об’єктів, для яких прийнято рішення про їх присутність на кадрі l_i усередині відповідного окаймляючого прямокутника.

Формується з елементів попередньої множини, тому структура кожного елементу така ж, як в

$$\forall b_d^{(i)''} \in B_j'', b_d^{(i)''} = \left(f_d^{(i)''} x_d^{(i)''} y_d^{(i)''} w_d^{(i)''} h_d^{(i)''} c_d^{(i)''} s_d^{(i)''} t_d^{(i)''}\right),$$
$$d = \overline{0, \left|B_j''\right| - 1},$$

$$T^i = \left\{ t_m^{(i)}, m = \overline{0, |B_i''| - 1} \right\}, i = \overline{0, N-1},$$

$$t_m^{(i)} = \left(b_{r_0}^{(fr_0)''}, b_{r_1}^{(fr_1)''}, \dots, b_{r_{s_{m-1}}}^{\left(f_{r_{s_{m-1}}}\right)''} \right), \qquad f_{r_{s_{m-1}}} \le i, \forall\, m = \overline{0, |T^\iota| - 1},$$

$$f_{r_0} < f_{r_1} < \dots < f_{r_{s_{m-1}}},$$

де $|T^i|$ – кількість траєкторій, побудованих до моменту обробки кадру l_i;

s_m – довжина траєкторії $t_m^{(i)}$.

Кожна траєкторія містить лише положення, для яких відомий ідентифікатор трека. *Tid* – останній вільний ідентифікатор траєкторій.

Нижче показана робоча множина для кожного кадру, що входить до складу блоку, який обробляється на ітерації методу. Вхідна множина показана в рамках, що мають форму правих стрілок, вихідні – в рамках, що мають форму лівих стрілок (рис. 2.6).

Визначившись з необхідними позначеннями, вхідними і вихідними даними однієї ітерації, перейдемо до опису обчислювальної схеми методу, запропонованого в рамках дослідження, яка застосовується до блоку зображень $l_i,\ l_{i+1}, l_i, \dots, l_{i+step}$.

Спочатку необхідно визначити множину положень об’єктів B_{i+step}, виконавши пошук об’єктів на зображенні l_{i+step}. Далі слід зіставити множину положень B_i та B_i' з побудованим набором B_{i+step}. Таке зіставлення дозволить отримати початкові відрізки нових траєкторій і продовжити вже існуючі траєкторії на етапі оновлення робочих множин. Потім потрібно виконати ряд допоміжних операцій (збереження набору положень, завантаження нового блоку кадрів і установка нинішніх множин), які дозволять перейти до розгляду наступного блоку кадрів.

Мета зіставлення – побудувати сукупності відповідностей MBS_j та MB_j положень B_j та B'_j з набором B_{i+step}, що задовольняють умові.

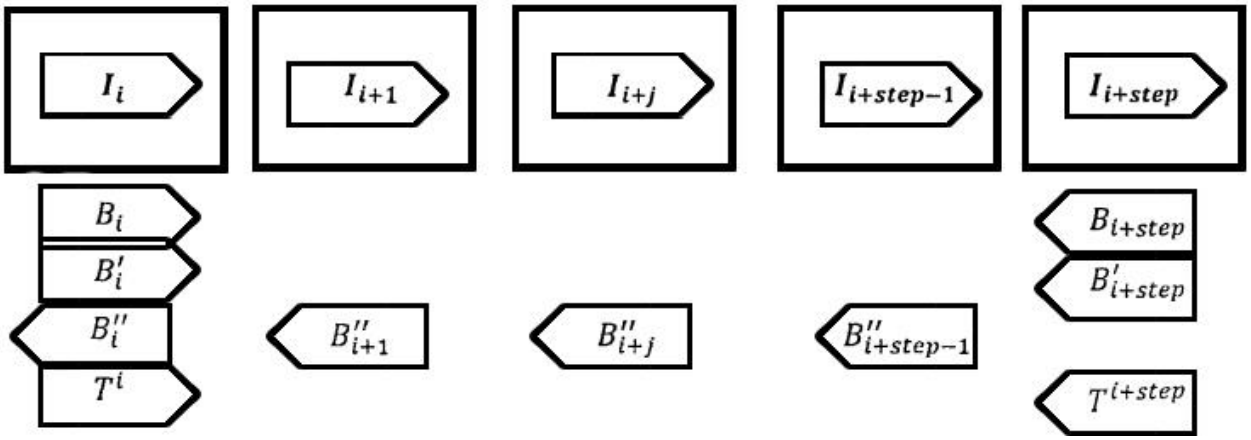

Рисунок 2.6 Робоча множина на кожному кадрі з блоку зображень

$$MBS_i = \begin{Bmatrix} (u, v_u, km_u), u = \overline{0, |B'_i| - 1}, v_u \in \{-1,0, \dots, |B_{i+step}| - 1\}, \\ \forall u_1, u_2 \in \{0, \dots, |B'_i| - 1\} / \ u_1 \neq u_2, v_{u_1} \neq v_{u_2} \end{Bmatrix},$$

$$MB_i = \begin{Bmatrix} (u, v_u, km_u), u = \overline{0, |B_i| - 1}, v_u \in \{-1,0, \dots, |B_{i+step}| - 1\}, \\ \forall u_1, u_2 \in \{0, \dots, |B_i| - 1\} / \ u_1 \neq u_2, v_{u_1} \neq v_{u_2} \end{Bmatrix},$$

$$\forall (u_1, v_{u_1}, km_{u_1}) \in MBS_i, \forall (u_2, v_{u_2}, km_{u_2}) \in MB_i \ v_{u_1} \neq v_{u_2},$$

де $v_u = -1$, якщо для положення з номером *u* не було знайдено відповідність; $km_u \in [0,1]$ – відносна кількість коректних відповідностей для конкретної пари положень, яке визначається як відношення числа коректних відповідностей до кількості ключових крапок, побудованих на частини окаймляючого зображення та обмежене прямокутником з *u*.

Зіставлення зводиться до перебору всіляких пар положень з множин $B_i \cup$ $\cup B'_i$ та B_{i+step}. В процесі пошуку вибирається положення $b_{s'}^{(i+step)}$, що відповідає максимальній відносній кількості коректних відповідностей ключових точок (рис. 2.7). Відзначимо, що така схема зіставлення не гарантує,

що елементи набору B_i та B_i' відображуватимуться в різні елементи множини B_{i+step}.

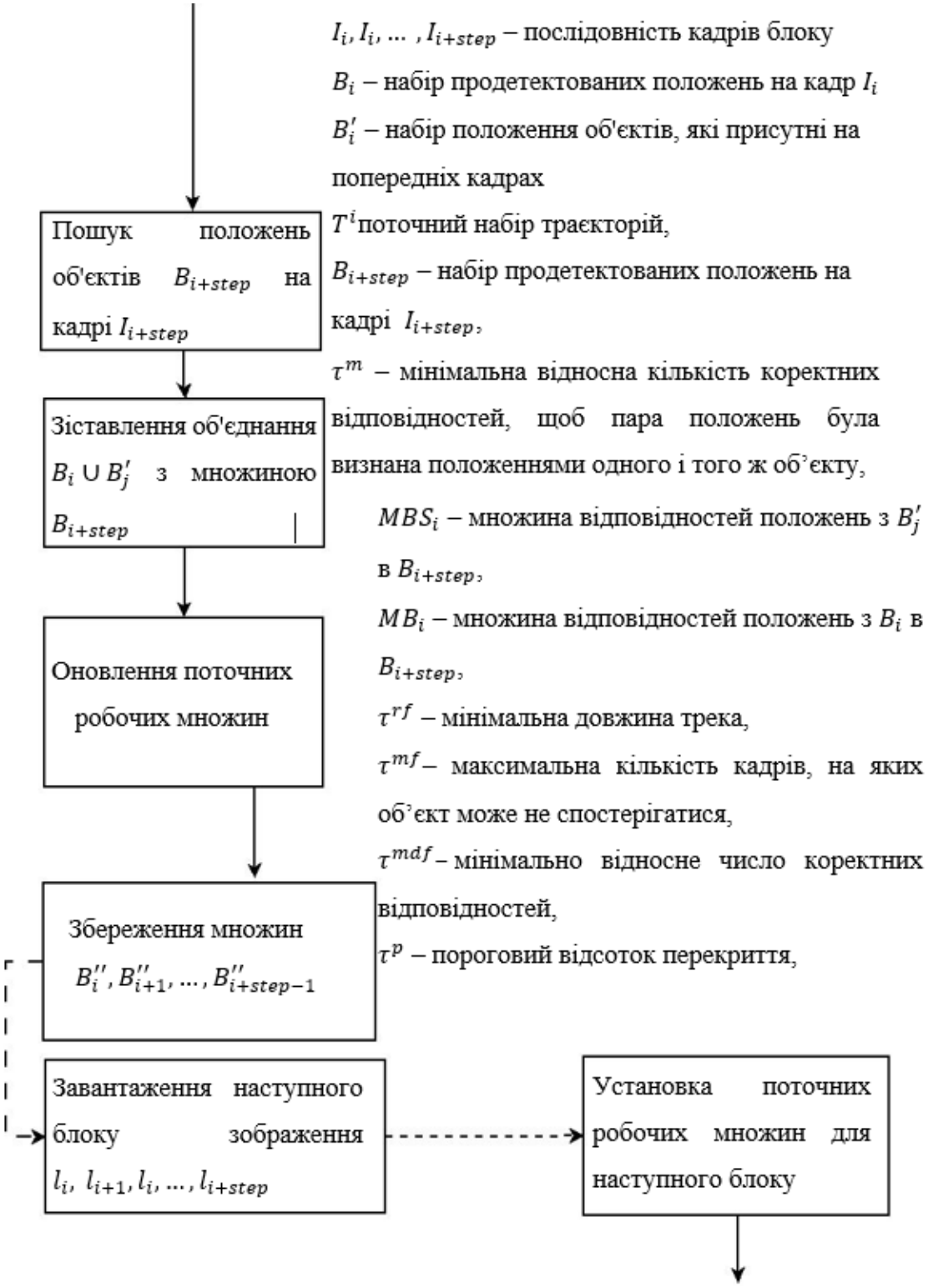

Рисунок 2.7 – Схема однієї ітерац ії алгоритму детектування з врахуванням результатів зіставлення

На підставі результатів зіставлення виконується оновлення робочих множин $B_i'', B_{i+1}'', \dots, B_{i+step}'', B_{i+step}', B_{i+step}$ і стани набору траєкторій T^{i+step}. Оновлення включає наступні кроки:

1) Оновлення на підставі множини відповідностей MBS_i. На даному кроці добудовуються продовження вже існуючих траєкторій (рис. 2.8) за допомогою відновлення положень з $b_{j_1}^{(i)'} \in B_j', \forall j_1 = 0\overline{\left|B_j'\right| - 1}$ на проміжних кадрах. Відмітимо, що, якщо для положення $b_{j_1}^{(i)'}$ знайдена відповідність $b_{j_{s_1}}^{(i+step)}$, то виконується контроль перекриття відновленого і продетектованого положень на останньому кадрі блоку. Відновленим положенням встановлюється ідентифікатор траєкторії $b_{j_1}^{(i)'}$ і приймається рішення про присутність об'єкту усередині відповідних окаймляючих прямокутників, тобто положення переміщаються в множини $B_i'', B_{i+1}'', \dots, B_{i+step}'', B_{i+step}', B_{i+step}$. Елементи множини B_{i+step}, для яких є прообрази в B_j', видаляються з B_{i+step}.

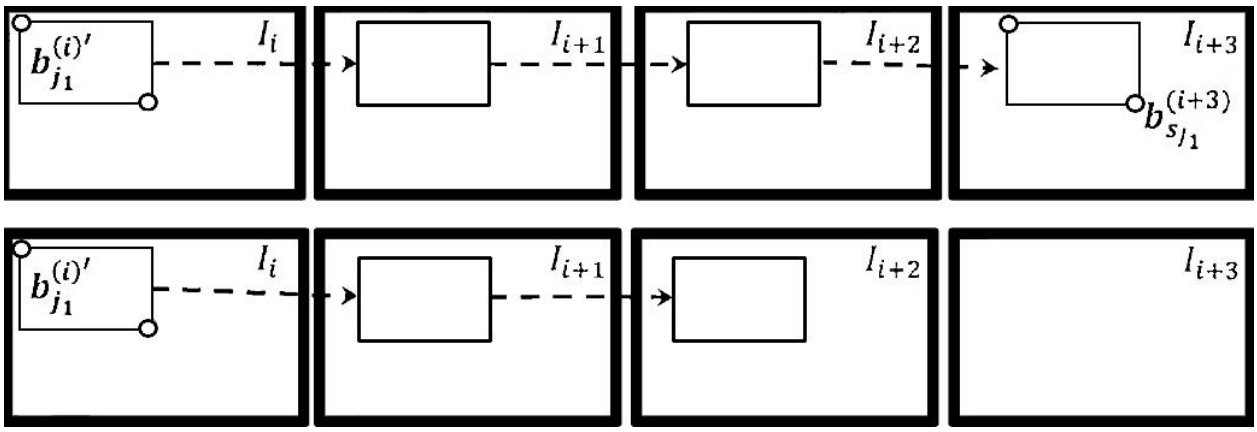

Рисунок 2.8 – Відновлення продовжень траєкторій для випадку *step = 3*

2) Оновлення на підставі множини відповідностей MB_j. У результаті виконання цього кроку формуються нові траєкторії, або оновлюються існуючі при умови, що об'єкт був поза областю огляду камери протягом деякого проміжку часу. За аналогією з попереднім кроком спочатку здійснюється

відновлення положень на проміжних кадрах. Далі виконується пошук серед поточного натовпу така траєкторія, для якої відновлена послідовність положень є продовженням. Пошук виконується за допомогою зіставлення кінцевих положень розглянутої траєкторії і початкових положень відновленої послідовності. Якщо таку траєкторію знайдено, то набір відновлених положень приєднується до кінця даної траєкторії з відповідним ідентифікатором. Якщо траєкторію не знайдено, а положення відновлено на всіх кадрах поточного блоку зображень, їх присвоюють останньому вільний ідентифікатор траєкторій *tid*, як наслідок, формується нова траєкторія.

Множини $B_i'', B_{i+1}'', \dots, B_{i+step}'', B_{i+step}', B_{i+step}$ доповнюються побудованими положеннями, при необхідності в поточний набір траєкторій T^i додається нова траєкторія. елементи множини B_{i+step}, для якої є прообрази в B_i також віддаляються з B_{i+step}.

3) Оновлення відповідно до поточного стану B_{i+step}. Після виконання перших двох кроків множина B_{i+step} містить лише положення, для яких не знайшлося прообразів в $B_i \cup B_j'$, це означає, що початок траєкторії знаходиться на одному з проміжних кадрів поточного блоку зображень. Тому виконується відновлення положень в зворотному часі (рис. 2.9). Потім по аналогії з попереднім кроком здійснюється пошук траєкторії, для якої відновлена послідовність може бути продовженням, і оновлення робочої множини $B_i'', B_{i+1}'', \dots, B_{i+step-1}'', B_{i+step}'$.

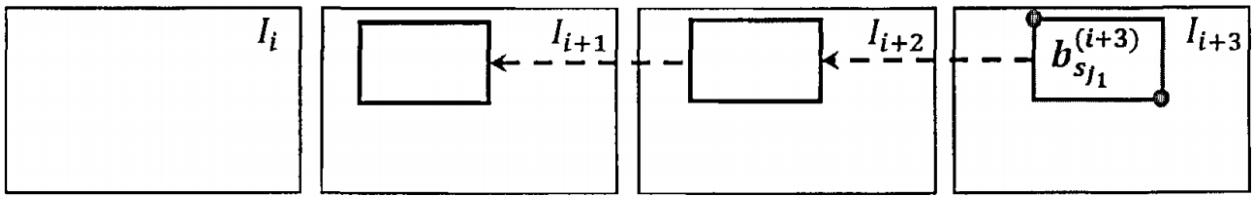

Рисунок 2.9 – Відновлення положень в зворотному часі для випадку *step=3*

Описаний алгоритм застосовується послідовно до кожного блоку кадрів відео. Зупинимося на етапі початкової ініціалізації робочої множини.

Очевидно, що множина B_0 формується в результаті запуску алгоритму детектування на кадрі l_0. Вважатимемо, що на початковому кадрі всі об'єкти виявлені правильно. Тому кожному побудованому положенню можна привласнити унікальний індекс траєкторії і перенести всі такі положення з множини B_0 в B_0'. В результаті $B_0 = \emptyset$ та $tid = |B_0'|$. Остання множина в початковий момент має бути порожньою: $B_0'' = B_1'' = \cdots = B_{step-1}'' = \emptyset$, $B_{step}' = \emptyset$. Поки не поступило жодного кадру логічно сукупність траєкторій також зробити порожньою $T^0 = \emptyset$.

Необхідно відзначити, що після закінчення обробки всього відео можна виконати додаткову обробку поста, пов'язану з видаленням невеликих траєкторій і, як наслідок, корекцією отриманого набору положень.

Розроблений метод багато в чому спирається на відновлення траєкторій з корекцією результатів відеодетектування. Вимога видимості достатньої кількості ключових точок також має місце.

3 ВИБІР ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ДЛЯ СТВОРЕННЯ СИСТЕМИ ВІДЕОДЕТЕКТУВАННЯ ТРАНСПОРТНИХ ЗАСОБІВ

3.1 Вибір інструментальних засобів для створення системи відеодетектування транспортних засобів

В процесі реалізації були використані наступні технології і програмні засоби:

– мови програмування: C++ використаний для реалізації основних модулів і додатків візуалізації; C#, Python, Matlab – для автоматизації збору і аналізу результатів відеодетектування;

– бібліотеки: OPENCV 2.4 містить відкриті реалізації базових алгоритмів комп'ютерного зору (Surf–деськріпторі, метод Latent SVM); Intel® Threading Building Blocks забезпечує конструкції для розпаралелювання програм для систем із загальною пам'яттю;

– ОС: Microsoft® Windows 7;

– середовища розробки: Microsoft® Visual Studio 2010, MATLAB R2011;

– система контролю версій: Tortoise SVN;

– інструмент побудови діаграм: Visual Paradigm for UML 7.0 (Community Edition).

До складу системи входить декілька програмних додатків і програм, які поділені на 5 груп відповідно до загальної архітектури:

1. Додатки перевірки якості пошуку та супроводу об'єктів:

   – ComputeDetectionRate, ComputeDetectionRateVisPercentage, ComputeDetectionRateVisPercentage2 – програма для оцінки якості пошуку;

   – ComputeAverageTrackingAccuracy – програма для оцінки якості супроводу об'єктів.

2. Утиліти для візуалізації:

   – PlayLayout – утиліта візуалізації положень об'єктів;

– PlayTracks – утиліта візуалізації траєкторій руху;

– PlayLayoutAndDetections – утиліта для одночасної візуалізації розмічених і виявлених окаймаючих прямокутників.

3. Допоміжні модулі та утиліти:

– Auxiliaries – бібліотека, що містить оголошення констант, реалізацію функцій читання/запису розмітки у файл, набір класів для уніфікації вилучення кадрів з відео і з послідовності зображень;

– MakeLayout – додаток для розмітки транспортних засобів на тестовому відео, початкова розмітка на кадрі виходить в результаті запуску алгоритму Latent SVM;

– MakeLayout2 – додаток для розмітки транспортних засобів на тестовому відео, початкова розмітка на кадрі зчитується з вхідного файлу;

– MakeLayout3_AddVisPerc – додаток для розмітки відсотка видимості об’єкта протягом всієї траєкторії руху;

– ROiLayout – додаток для розмітки областей інтересу;

– CreceOverlappinglmagesSet – програма для створення набору перекритих зображень.

4. Набір бібліотек, що містять логіку:

– DetectionModule – бібліотека пошуку об’єктів;

– TrackingModule – бібліотека функцій, необхідних для супроводу об’єктів і зіставлення зображень об’єктів.

5. Набір реалізацій методів відеодетектування:

– FullDetection – додаток, що містить реалізацію схеми повного покадрового детектування транспортних засобів різних класів на відео. Можливість відрегульовано параметрами програми;

– DetectionTracking – програма, що містить реалізацію методу, який поєднує детектування і супровід. Крок, з яким виконується чергова операція пошуку, визначається вхідними параметрами програми;

– CreceFullTracks – реалізація алгоритму відновлення траєкторій руху з корекцією результатів відеодетектування;

– CreceFullTracksOnline – реалізація методу детектування, що враховує результати зіставлення.

Зупинимося більш докладно на ключових функціях і класах, розроблених в кожному модулі, і коротко розглянемо схеми їх взаємодії. Перші три групи модулів і утиліт, за винятком Auxiliaries, є самостійними і не використовують засоби інших бібліотек, що входять до складу системи.

Допоміжна бібліотека Auxiliaries серед усього іншого містить набір класів, забезпечують уніфікацію роботи з кадрами відео і сукупністю зображень (рис. 3.1). До складу входить абстрактний клас GrabFrame, який являє собою інтерфейс доступу до даних і включає реалізацію трьох методів: отримання чергового кадру (grabFrame), перехід на початок послідовності кадрів (resetGrabber), перехід на вказаний кадр послідовності (seekGrabber). Також є дві реалізації цього інтерфейсу для випадку відео (CaptureGrabFrame) і для випадки послідовності зображень (SequenceGrabFrame). Цей набір класів використовується далі у всіх запропонованих схемах відеодетектування.

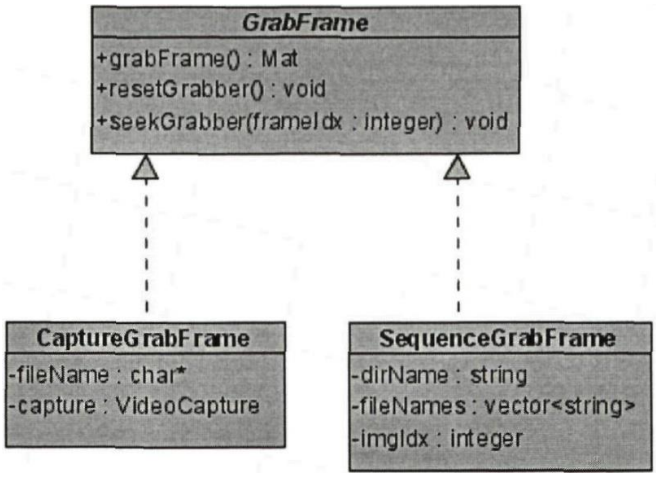

Рисунок 3.1 – Схема спадкоємства класів для уніфікації роботи з кадрами відео і послідовністю зображень

Бібліотеки DetectionModule і TrackingModule містять функції, необхідні для реалізації різних схем відеодетектування. DetectionModule складається з двох програмних додатків:

– object_detection – модуль класів, що забезпечують пошук об’єктів на кадрі як на самостійному зображенні (рис. 3.2). До складу входить абстрактний клас VehicleDetector і його реалізація, що використовує алгоритм LatentSVM (vehicleDetectorLSVM). Передбачено методи пошуку без відсікання по областях інтересу та з відсіканням (DetectObjects);

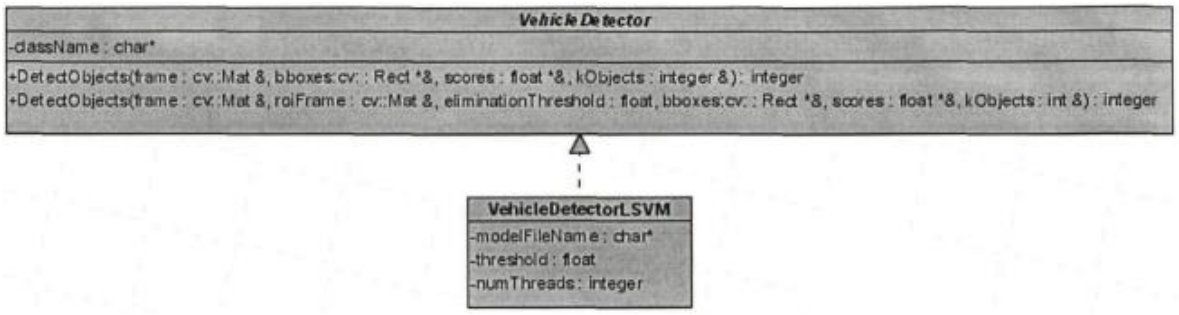

Рисунок 3.2 – Схема спадкоємства класів для пошуку транспортних засобів різних класів

– Bboxes_elimination – модуль функцій, що забезпечують відкидання множини виявлених положень відповідно до областей інтересу. Додаток містить одну ключову функцію truncceBoundingBoxesList, яка повертає набір індексів положень, що виходять за межі області інтересу більше, ніж на задане порогове значення.

Бібліотека TrackingModule складається з чотирьох програмних додатків:

– bboxes_tracking містить функції, необхідні для реалізації схеми з супроводом, заснованим на застосуванні алгоритму Лукаса-Канаді: обчислення дескрипторів для супроводу (getBoundingBoxesFeatures), супровід набору дескрипторів (computeOpticalFlowLC), відновлення положення прямокутника на заснування розташування дескрипторів на попередньому і поточному зображеннях (restoreBoundingBoxesLocation), суміщення положень, отриманих у результаті супроводу та детектування (combceDetectedTrackedBBoxes);

– bboxes_matching містить клас, що забезпечує можливість зіставлення пари зображень prevImg і currImg (рис. 3.3) з використанням детектора

ключових точок featureDetector, дескриптора descriptorExtractor і методу зіставлення descriptorMatcher, які мають типи, визначені в бібліотеці OPENCV.

У класі реалізовано методи для зіставлення зображень (Match), отримання першої коректної відповідності ключових точок (GetFirstinlier), отримання кількості коректних відповідностей ключових точок (GetNumMatches), відображення дескрипторів VideoCapture (ShowMatches) на зображенні та очищення робочих множин (clearMatches). У процесі порівняння (виклик методу Match) набори ключових точок зберігаються у множині keyPointsPrevImg і keyPointsCurrlmg, а відповідні пари у вектор matches, з якого виводяться коректні пари за допомогою методу RANSAC і формується множина коректних відповідностей inliers;

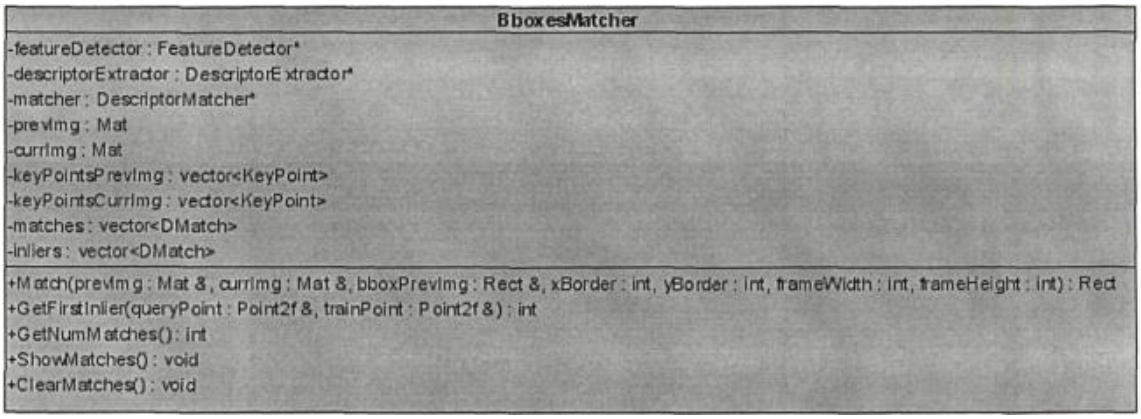

Рисунок 3.3 – Поля і методи класу Bboxesmatcher

– online_tracks_creator складається з класів і функцій, необхідних для реалізації незалежного алгоритму корекції результатів детектування в процесі відновлення траєкторій руху об’єктів (рис. 3.4). Відповідно до опису даного алгоритму модуль містить три абстракції для представлення виявленого положення об’єкту (клас Record), положення об’єкту, яке належить деякій траєкторії (клас Trackrecord), і для представлення самої траєкторії (Track). У складі модуля є набір функцій для зіставлення положень об’єктів (matchrecords), відновлення «пропущених» положень в траєкторії (restorerecords), розбиття множини продетектованих положень на треки

(splitbboxes2tracks), злиття траєкторій (mergetracks, merge2tracks), відкидання невеликих траєкторій (rejectshorttracks). Вказані функції використовуються при реалізації алгоритму відновлення траєкторій з корекцією результатів відеодетектування;

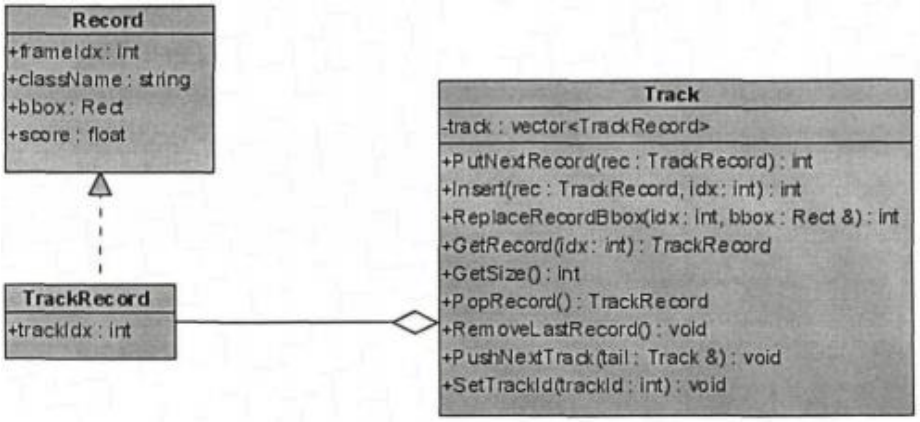

Рисунок 3.4 – Класи для представлення положення об’єкту і траєкторії його руху

– FullDetection використовує функціонал бібліотек Auxiliaries і DetectionModule;

– DetectionTracking використовує функціонал бібліотек Auxiliaries, DetectionModule і TrackingModule (модуль bboxes_tracking);

– CreateFullTracks використовує функціонал бібліотек Auxiliaries, TrackingModule (модуль tracks_creator);

– CreateFullTracksOnline використовує функціонал бібліотек Auxiliaries, DetectionModule I TrackingModule (модуль online_tracks_creator).

У цьому параграфі описані програмні бібліотеки та модулі, які мають безпосереднє відношення до реалізації розроблених методів детектування. Поряд з цим, система містить набір допоміжних програм, що забезпечують аналіз і підготовку тренувальних даних для класифікатора Latent SVM, серед яких можна виділити ряд основних:

– CarsNonCarsCorrelation – додаток для визначення співвідношення числа положень у всьому наборі зображень, що містять об’єкти заданого класу, до кількості положень їх не містять;

– CarsNonCarsTrainValCorrelation – додаток, що дозволяє визначити співвідношення числа положень в тренувальній і валідаційної вибірці, що містять об’єкти заданого класу, до кількості положень їх не містять;

– CreateTrainData – додаток для створення тренувальної бази для класифікатора Latent SVM в форматі, наданому розробниками алгоритму;

– ChooseTrainDetections – додаток для вибору положень транспортних «коштів заданого класу, на яких буде навчатися класифікатор Latent SVM;

– GetTrackEmptyFrames – додаток для вибору положень, які не містять об’єкти заданого класу для подальшого навчання класифікатора Latent SVM.

Також є набір додатків, що дозволяють автоматизацію процес аналізу параметрів методів.

CreateOverlappinglmgsFeaturesTable – додаток для побудови таблиці, яка містить число видимих ключових точок, що належать об’єкту, в залежності від типу і відсотка перекриття фоновими зображенням. Розглядається 12 типів перекриття і відсоток перекриття змінюється від 10% до 90% з кроком 10%, сформованих за допомогою програми CreateOverlappinglmagesSet.

3.2 Структурна схема та опис модулів системи відеодетектування транспортних засобів

Модулі пошуку, супроводу та зіставлення [11] є центральними в розробленої програмної системі. На базі функціоналу зазначених модулів будується реалізація методів детектування, зокрема, розглянутих і запропонованих в рамках наукового дослідження: методу повного покадрового пошуку, методу повного покадрового пошуку з відсіканням по областям інтересу, методу детектування з урахуванням результатів супроводу (на один і кілька кадрів, методу детектування для відновлення траєкторій руху,

методу детектування, що забезпечує пошук на деяких кадрах відео потоку, відновлення положень на проміжних кадрах і одночасне побудова траєкторій.

В процесі пошуку об’єктів в кожному методі присутній можливість введення видалення окаймляючих прямокутників, що виходять з областей інтересу. Розмітка областей інтересу здійснюється з використанням відповідного модуля системи в припущенні, що відео отримано з нерухомою камери (допустимі незначні зміщення в результаті зміни погодних умов, наприклад, дощу або вітру). Результати детектування можна візуалізувати в режимі перегляду відео за допомогою модулів візуалізації окаймляючих прямокутників і траєкторій руху об’єктів.

Аналіз якості роботи системи детектування вимагає введення модулів підготовки тестових даних – модулі розмітки окаймляють прямокутники для транспортних засобів різних класів, модулі формування траєкторій їх руху і модулі розмітки відсотка видимості об’єкта на підставі розмітки положень і траєкторій. Тестове відео з розміченими прямокутниками також може бути визуализировано. Запуск реалізації будь-якого методу детектування дозволить отримати набір положень і траєкторій, які в процесі аналізу якості необхідно порівняти з розміткою за допомогою деяких кількісних показників. модулі апробації результатів якості детектування надають відповідний функціонал, а допоміжні утиліти дозволяють автоматизувати процес збору та систематизації показників.

Модулі супроводу і зіставлення. Модуль супроводу [12] дозволяє відновити становище на наступному кадрі для кожного продетектированного на поточному кадрі окаймляючого прямокутника. Таким чином, в якості вхідних параметрів модуль приймає пару послідовних кадрів відео, набір виявлених окаймляючих прямокутників на першому кадрі і сукупність параметрів алгоритму супроводження. На виході формується послідовність окаймляючих прямокутників, відповідних вхідного набору прямокутників.

Реалізація модуля передбачає наступну схему роботи:

1. Виділення підобластей поточного зображення, які обмежені виявленими окаймляючими прямокутниками.

2. Пошук особливих точок для подальшого супроводу в кожної підобласті.

3. Визначення положення множини будь-яких спеціальних точок на наступному кадрі відео з допомогою алгоритму Лукаса-Канаді.

4. Відновлення положень окаймляючих прямокутників на наступному кадрі відео:

– пошук особливої точки на наступному кадрі, яка не була втрачена в результаті супроводу;

– визначення положення окаймляющего прямокутника щодо відповідної точки на поточному кадрі;

– обчислення координат окаймляющего прямокутника на наступному кадрі в припущенні, що розподіл особливих точок при переході від поточного кадру до наступного не змінюється.

Модуль зіставлення також дозволяє відновити становище об’єкта на наступному / попередньому кадрі послідовності.

Реалізація модуля включає:

1. Можливість зіставлення пари зображень для перевірки того, що це зображень одного і того ж об’єкта. Зіставлення включає наступні дії:

– виділення ключових точок з використанням SURF-детектора (передбачена можливість застосування інших детекторів);

– обчислення SURF-дескрипторів ключових точок (передбачена можливість вибору інших дескрипторів);

– «грубе» зіставлення дескрипторів і відсікання помилкових спрацьовувань з допомогою алгоритму RANSAC.

2. Функції, необхідні для реалізації алгоритму відновлення траєкторій з корекцією результатів детектування:

– функції розбиття множини виявлених положень на треки;

– функції склеювання треків в траєкторії;

– функції відновлення пропущених положень в траєкторіях.

3. Функції, необхідні для реалізації схеми детектування, що враховує результати зіставлення.

Базові операції роботи з зображеннями, супровід особливих точок, обчислення і зіставлення дескрипторів виконується засобами бібліотеки OpenCV.

Модуль розмітки областей інтересу [13]. Модуль розмітки областей інтересу надає можливість відсікти області зображення, в якому разі не можуть потрапити транспортні засоби, наприклад, узбіччя оглядає камерою дорожнього ділянки. На вході модуль приймає будь-який кадр відео з дорожнім рухом або саме відео. В результаті користувальницької розмітки на виході формується бінарне зображення, на якому області білого кольору відповідають областям інтересу, а чорні – областям відсікання.

Робота програми включає наступні етапи:

1. Завантаження вхідного зображення або першого кадру відео.

2. Вибіркова розмітка областей інтересу за допомогою простого зафарбовування областей відсікання мишею. Кисть реалізована у вигляді кола, радіус якої можна змінити, зміщуючи повзунок интерфейсного компонента. розподіл шкали компонента відповідає радіусу кисті.

Модуль розмітки окаймляючих прямокутників [14]. Для автоматизації процедури розмітки транспортних засобів на відео потоці розроблені відповідний модуль і візуальне додаток. Додаток в якості вхідних параметрів приймає назву відео, на якому буде виконуватися розмітка, розмітка областей інтересу і перелік розмічаємо класів транспортних засобів. Результатом роботи програми є послідовність окаймляючих прямокутників, розмічених на кожному кадрі вхідного відео. Передбачена наступна схема роботи програми:

1. Детектор знаходить приблизне розташування транспортних засобів можливих класів на поточному кадрі. Кожному класу призначається свій колір окаймляючих прямокутників. Для розрізнення класів поряд з основним вікном розмітки є палітра класів. Палітра класів транспортних засобів.

2. Користувач при необхідності виправляє розташування прямокутників. Передбачена можливість зсуву сторін, видалення прямокутника і відтворення нового прямокутника. Попередньо потрібно вибрати редагований клас об’єктів на палітре.

3. При переході на наступний кадр відео забезпечується збереження побудованих прямокутників в файл в форматі:

<frame_idx> <class> <x_1> <y_1> <x_2> <y_2>,

де *<frame_idx>* – індекс кадру в послідовності кадрів вхідного відео;

<class> – назва класу транспортного засобу;

<x_1> <y_1> <x_2> <y_2> – координати лівого верхнього і правого нижнього кутів окаймляючого прямокутника.

Нижче наведена цитата з файлу розмітки.

571 CAR 586 25 614 50

571 BUS 506 8 552 56

Модуль формування траєкторій руху на тестовому відео. Підхід, прийнятий для розмітки траєкторій руху при наявності розмітки транспортних засобів полягає в тому, щоб відстежувати перетин окаймляючих прямокутників на поточному та наступному кадрі відеопотоку. Розроблена наступна схема розмітки траєкторій:

1) Вибирається окаймляющий прямокутник, який не був задіяний до теперішнього моменту ні в одній з траєкторій. Приймається, що це початок траєкторії. Фіксується кадр, на якому був розмічений цей прямокутник.

2) Забезпечується перегляд послідовності наступних кадрів і для кожного виконується пошук прямокутника, що має площу перетину з прямокутником, який є останнім у траєкторії, більше заданого порогу. Якщо такий прямокутник знайшовся, то він приймається за поточний кінець траєкторії. Процес триває до тих пір, поки на черговому кадрі не буде

знайдений відповідний прямокутник. При необхідності поріг можна варіювати.

3) Збереження траєкторії. Прийнято наступний формат зберігання траєкторій руху:

$$<Len> <idx_l> <idx_2> \cdots <idx_len>,$$

де <*len*> – довжина траєкторії;

<*idx*> – індекс розміченого окаймляющего прямокутника в файлі розмітки.

4) Попередні кроки повторюються, поки для кожного розміченого окаймляючого прямокутника не буде визначено траєкторія. Результат розмітки траєкторій на відео можна переглянути в динаміці за допомогою модуля візуалізації траєкторій, використовуючи отриманий набір траєкторій і розмітку окаймляючих прямокутників.

Відзначимо, що в процесі застосування даної схеми наявність перекриття об’єктів в деяких випадках призводить до розриву траєкторії. Тому можна запустити реалізацію алгоритму відновлення траєкторій з корекцією результатів детектування на множині розмічених положень. Причому для відновлення траєкторій досить застосувати етап розбиття множини окаймляючих прямокутників на короткі траєкторії з подальшим склеюванням невеликих треків без відновлення положень на проміжних кадрах. Експерименти показали, що застосування даного алгоритму до тестового відео у 2000 кадрів, який містить приблизно 3000 положень, середня точність супроводу склала 0,928, а час обробки відео – приблизно 8,5 хвилин. Втрата точності супроводу пояснюється тим, що на відео розмічені транспортні засоби в моменти появи / зникнення, коли відсоток видимості об’єкта не дозволяє виділити достатню кількість ключових точок. Тому потрібно вручну доповнити траєкторії відсутніми положеннями розмітки, скориставшись засобами візуалізації поточної положень і траєкторій об’єктів.

Модуль розмітки відсотка видимості об’єкту [15]. Модуль розмітки відсотка видимості дозволяє за заданими наборами окаймляючих прямокутників і траєкторій задати відсоток видимості об’єктів на кожному кадрі відео. При необхідності модуль надає можливість автоматично розмітити відсоток видимості на підставі заданих відсотків сусідніх положень.

Модуль включає наступні дії:

1. Завантаження зображень об’єкта відповідно до послідовності окаймляючих прямокутників, які становлять траєкторію руху.

2. Розміщення виділених зображень на одному зображенні і відображення траєкторії (рис. 3.5).

3. Виділення зображень об’єкта, для яких можливо оцінити відсоток видимості, і установка відповідного відсотка видимості.

4. Інтерполяція відсотка видимості для всіх сусідніх положень зліва і справа, поки не зустрінеться початок / кінець траєкторії або інше положення, для якого був встановлений відсоток видимості.

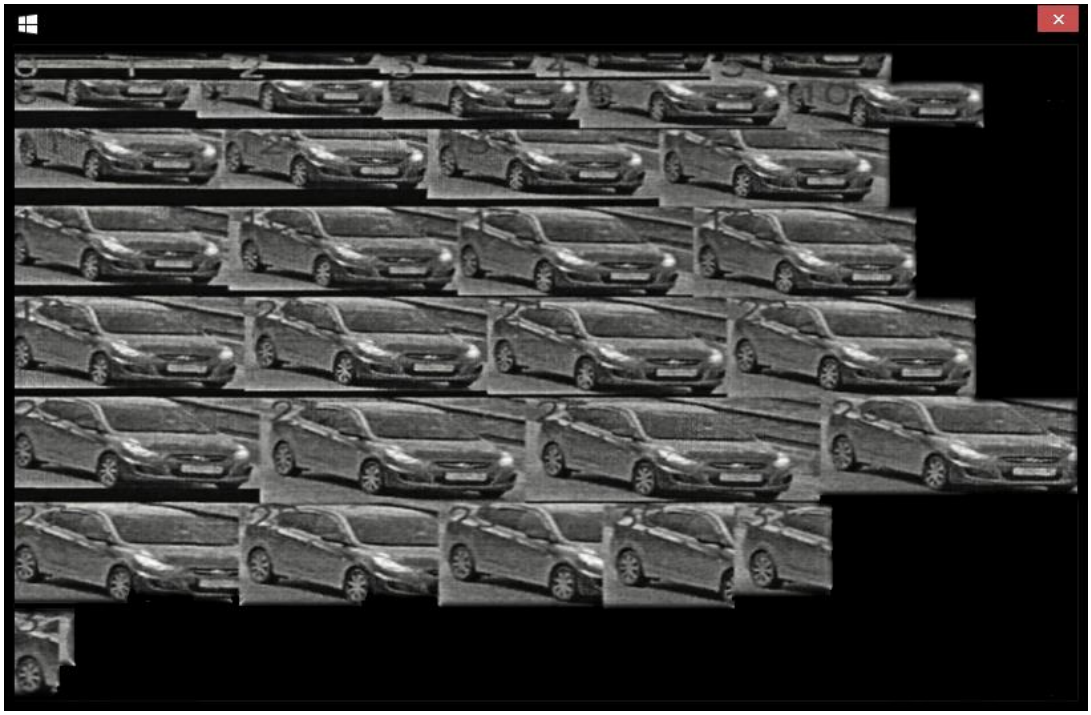

Рисунок 3.5 – Приклад відображення траєкторії руху транспортного засобу

Модулі візуалізації окаймляючих прямокутників і траєкторій руху. Модулі візуалізації окаймляючих прямокутників і траєкторій руху призначені для наочного контролю якості детектування і створеної розмітки. Перший модуль виконує відображення розмічених або виявлених прямокутників на заданому відео. Другий – забезпечує відображення розмічених або побудованих траєкторій руху транспортних засобів. Відповідні додатки можуть бути запущені як в режимі перегляду відео, так і в режимі ручного перегортання кадрів.

Схема роботи модуля візуалізації окаймляючих прямокутників включає завантаження поточного кадру відео і відповідного набору окаймляють отрисовку прямокутників на поточному кадрі і відображення поточного кадру з подальшим переходом до розгляду наступного кадру [16].

Для візуалізації траєкторій руху досить поставити у відповідність кожному розміченому прямокутника ідентифікатор траєкторії, тобто зіставити вміст файлу з окаймляючими прямокутниками і траєкторіями. У зв’язку з цим в модулі реалізована наступна послідовність дій:

1. Завантаження окаймляючих прямокутників.
2. Завантаження траєкторій і установка власних квітів для кожної траєкторії руху.
3. Перегляд набору індексів окаймляючих прямокутників, що складають кожну траєкторію, і присвоювання індексу даної траєкторії відповідним окаймляючим прямокутникам.
4. Завантаження поточного кадру відео.
5. Отрісовка прямокутників на поточному кадрі з зазначенням індексу траєкторії. Колір прямокутника визначається кольором, встановленим для траєкторії.
6. Відображення поточного кадру.
7. Перехід до наступного кадру, повторне виконання попередніх трьох дій.

Модулі обчислення кількісних показників якості [17]. Модуль обчислення показників якості пошуку на підставі наборів розмічених і виявлених окаймляючих прямокутників забезпечує обчислення наступних метрик:

– кількість правильно виявлених об’єктів (true positives);

– число прямокутників, в яких детектор виявив об’єкт, хоча в дійсності його там немає (false positives);

– кількість розмічених окаймляючих прямокутників;

– загальна кількість кадрів в потоці.

Відзначимо, що при обчисленні зазначених показників порогове значення відсотка перетину є вхідним параметром. Результуючі значення згодом використовуються засобами автоматизації для обчислення ряду показників якості пошуку.

Модуль обчислення показників якості супроводу забезпечує визначення середньої точності супроводу за допомогою зіставлення розмічених і побудованих траєкторій.

Засоби автоматизації для збору результатів оцінювання якості детектування [18]. Засоби автоматизації включають набір скриптів (Python, Matlab), що забезпечують визначення показників якості пошуку (TPR, FDR, AverFPperFrame, AverTPperFrame, AverFPperObject, AP) і супроводу об’єктів для результатів, отриманих при запуску реалізацій різних методів детектування на тестових відео.

Основний скрипт отримує набір директорій, що містять файли з виявленими окаймляющими прямокутниками і побудованими траєкторіями, а також список файлів з еталонною розміткою положень і траєкторій для відповідних тестових відео.

Скрипт виконує прохід по всіх тек, в яких знаходяться результати детектування, і здійснює порівняння результатів з еталонною розміткою за допомогою використання модулів обчислення кількісних показників якості. Для інтеграції із зовнішнім модулем VOCdevkit, який реалізований на базі

пакету Matlab, розроблені допоміжні скрипти з використанням Matlab, що дозволяють зберігати precision / recall-криві і значення вказанної метрики якості.

3.3 Тестування розробленого програмного засобу

Для проведення експериментів і аналізу якості розроблених методів було знято кілька відео (частота – 25 FPS, дозвіл – 720×405 пікселів). Необхідність підготовки власних тестових даних зумовлена тим, що в процесі виконання роботи не вдалося знайти загальноприйнятих баз дорожніх відео, які відповідають умовами спрацьовування алгоритму пошуку [19, 20].

Нижче приведена характеристика відзнятих тестових відео:

– track_09_0–2000 (2000 кадрів = 80 секунд, ~17000 окаймляючих прямокутників) містить велику кількість невеликих транспортних засобів трьох класів («АВТОМОБІЛЬ», «ТРАМВАЙ», «АВТОБУС»), що рухаються в двох протилежних напрямках. Характерна особливість даного відео полягає в тому, що значну площа кадру займає фон;

– track_10_5000–7000 (2000 кадрів = 80 секунд, ~3000 окаймляючих прямокутників, 58 траєкторій) – відео з транспортними засобами єдиного класу «АВТОМОБІЛЬ», які рухаються в 4 смуги одного напрямку;

– track_10_7000–8000 (1000 кадрів = 40 секунд, ~1000 окаймляючих прямокутників, 30 траєкторій). Містить об’єкти класів «АВТОМОБІЛЬ» і «АВТОБУС»;

– track_10_9000–11000 (2000 кадрів = 80 секунд, ~2300 окаймляючих прямокутників, 48 траєкторій). Присутні об’єкти класів «АВТОМОБІЛЬ» і «АВТОБУС».

Основна відмінність даного тестового відео – наявність великого числа вантажних транспортних засобів.

Нижче (рис. 3.6) показані окремі кадри тестових відео. Нескладно бачити, що принципова відмінність відео track_09_ *, track_10_ * – розмір об’єктів і наявність / відсутність зміни ракурсу в процесі руху.

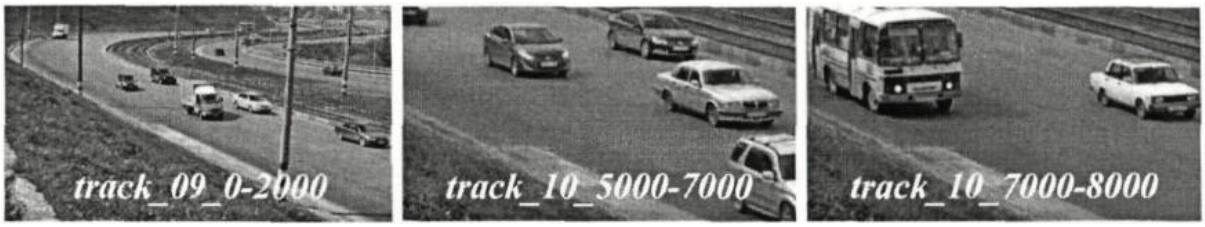

Рисунок 3.6 – Приклади кадрів з тестових відео

В процесі розмітки зазначалися положення всіх транспортних засобів, включаючи частково видимі об’єкти (до 2% видимості). По суті була виконана повна розмітка об’єктів на тестових відео.

Додатково були згенеровані синтетичні відеопослідовності довжина 2000 кадрів, що містять нерухомі транспортні засоби. Відеопослідовності отримані в результаті багаторазового копіювання одного кадру (рис. 3.7), що спрощує розмітку положень і траєкторій:

– track_10_5000–7000_l044×2000. Відео містить 3 об’єкти: 2 частково видимих, причому 1 бачимо менш ніж на 50% (легковий автомобіль);

– track_10_5000–70061192×2000. Містить 4 об’єкти: 3 повністю видимих, 1 видно більш ніж на 50% (легковий автомобіль, що входить в область огляду камерою);

– track_10_5000–7000_656×2000. Містить 3 об’єкти: 2 повністю видимих, 1 видно більш ніж на 50% (вантажний автомобіль, що входить в область огляду камерою, кабіна водія видно повністю);

– track_10_9000–11000_206×2000. Містить 2 повністю видимих об’єкта (вантажний і легковий автомобілі).

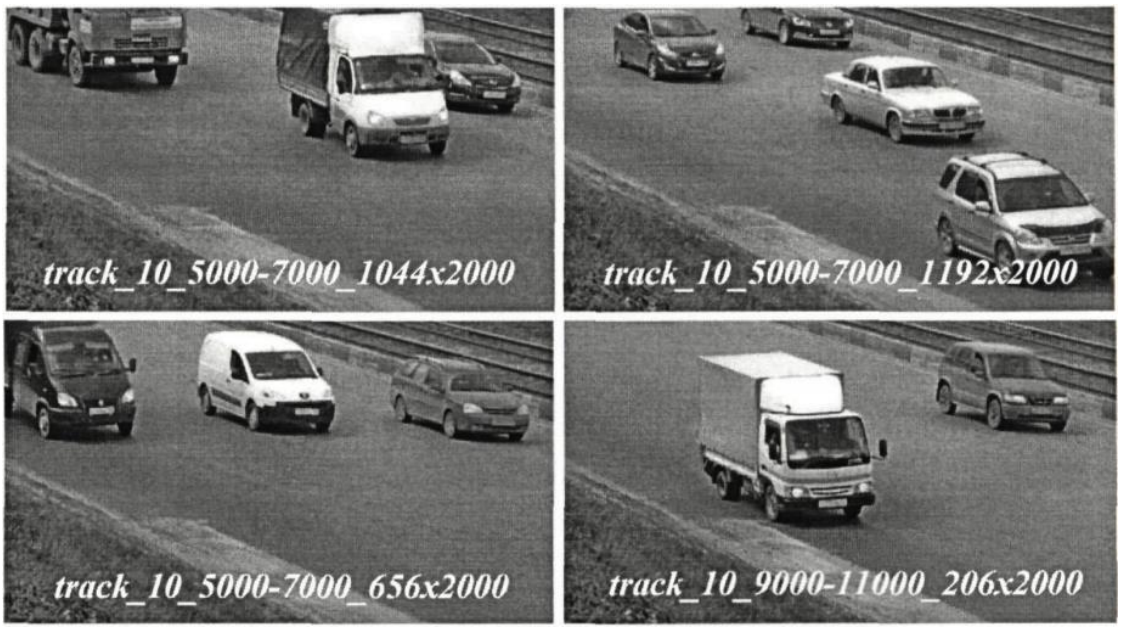

Рисунок 3.7 – Приклади кадрів з тестових відео

При аналізі якості пошуку необхідно розглядати положення тих об’єктів розмітки, які досить видимі, щоб алгоритм пошуку виявив об’єкт на зображенні. Тому попередньо слід оцінити мінімальний відсоток видимості транспортного засобу, при якому воно буде знайдено алгоритмом. Для визначення мінімального відсотка видимості необхідно провести серію експериментів і проаналізувати результати пошуку при різних відсотках перекриття об’єктів фоном. Виберемо зображення транспортних засобів, які мають принципово різний зовнішній вигляд. Далі для визначеності будемо розглядати клас «АВТОМОБІЛЬ» і об’єкти, показані нижче (рис 3.8). Сформуємо набори перекритих зображень з використанням розроблених допоміжних утиліт (CreateOverlappinglmagesSet). Типи перекриттів були приведені на потім запустимо реалізацію методу повного покадрового пошуку (FullDetection) для отриманого набору зображень і сформуємо з результатів таблицю (CreateOverlappinglmgsTable), яку згодом будемо аналізувати.

Рисунок 3.8 – Тестові зображення класу «АВТОМОБІЛЬ» для аналізумінімального відсотка видимості

Проведемо аналіз мінімального відсотка перекриття для алгоритму пошуку Latent SVM. Модель класу «АВТОМОБІЛЬ» отримана в результаті навчання Latent SVM на наборі зображень, що містять 1650 об'єктів даного ласу (50% з бази VOC 2007, 50% з track_10_5000–7000), і 4250 зображень, на яких відсутні об'єкти даного класу. Наступні три таблиці (рис. 3.9 – рис. 3.11) містять результати пошуку об'єктів класу «АВТОМОБІЛЬ» на наборі зображень, сформованих в процесі застосування різних перекриттів до тестових зображень 1, 2 і 3 відповідно (рис. 3.8).

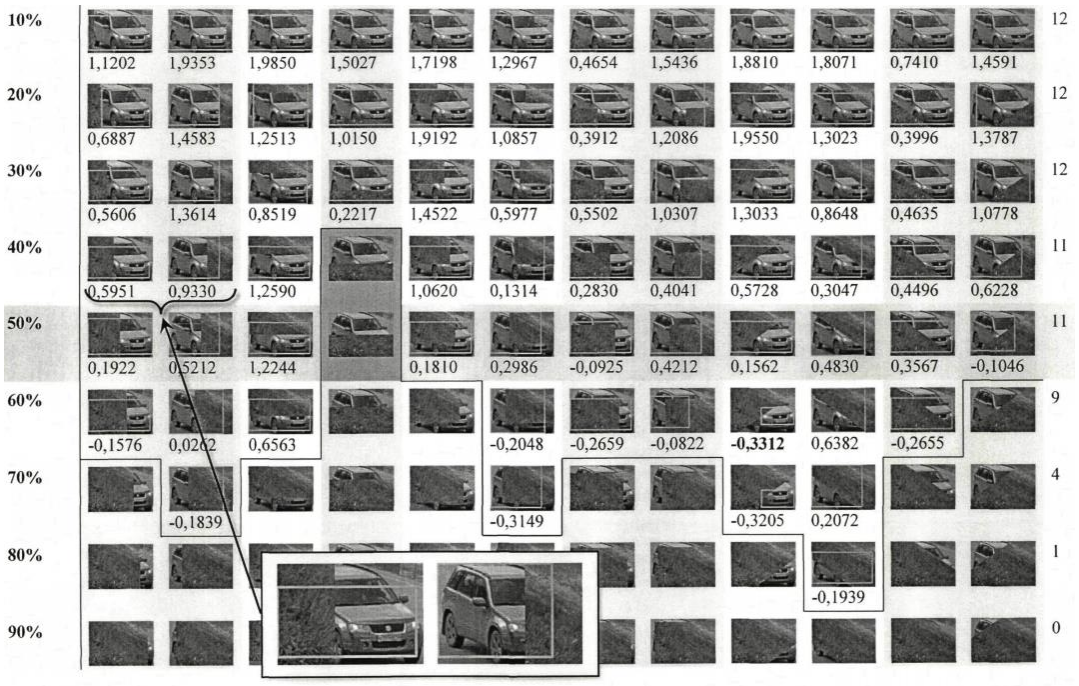

Рисунок 3.9 – Результати пошуку об'єктів класу «АВТОМОБІЛЬ» для набору перекриттів тестового зображення рис. 3.8

Рисунок 3.10 – Результати пошуку об’єктів класу «АВТОМОБІЛЬ» для набору перекриттів тестового зображення рис. 3.8

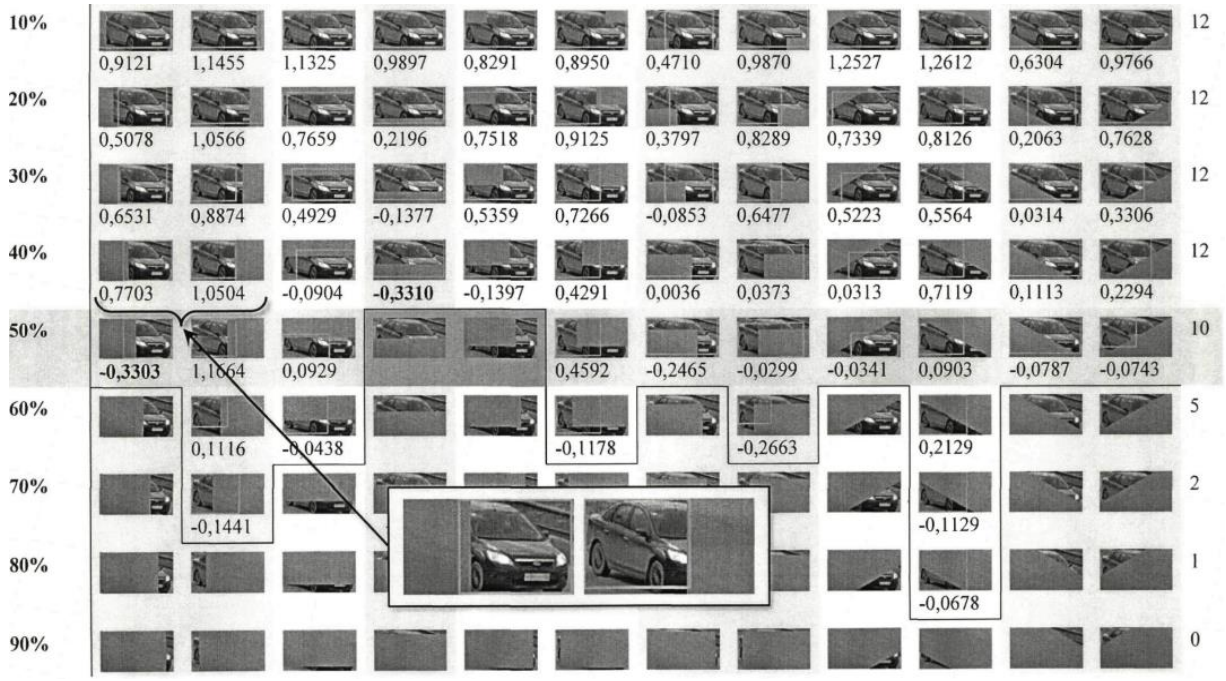

Рисунок 3.11 – Результати пошуку об’єктів класу «АВТОМОБІЛЬ» для набору перекриттів тестового зображення рис. 3.8

Перший стовпець таблиць відповідає значенню відсотка перекриття, останній – числу спрацьовувань алгоритму пошуку при заданому відсотку перекриття. Всі інші осередки містять перекрите зображення, на якому рамкою показаний окаймляющий прямокутник, також під зображенням наведено значення достовірності присутності об'єкта в даному прямокутнику. Відсутність значення достовірності в осередку означає, що алгоритм не виявлено об'єкт. Ламана лінія для кожного розглянутого типу перекриття розділяє набори зображень, де спрацював і не спрацював алгоритм пошуку.

Спочатку з безлічі тестованих типів перекриттів виберемо такі, які реально можуть виникати в процесі зйомки дорожнього руху в заданому ракурсі. Об'єкт знаходиться в ракурсі «три чверті», інші транспортні засоби рухаються сонаправленнимі тестовому об'єкту, тому природно можуть виникати перекриття його нижньої частини (затемнені стовпці 5, 8, 9, 12, 13). У разі якщо перекриваючий об'єкт знаходиться на передньому плані і має геометричні розміри, що значно перевищують розміри тестового транспортного засобу, то також можливі ліві і праві вертикальні перекриття (затемнені стовпці 2, 3).

Відзначимо, що в таблицях присутній невелика кількість положень, ступінь достовірності присутності в яких близька до мінімальної достовірності, отриманої в результаті навчання моделі (-0,332666).

З отриманих результатів не складно бачити, що вже при 50% видимості автомобілів алгоритм пошуку Latent SVM, навчений на зазначеному раніше наборі даних, спрацьовує не на всіх типах перекриттів. Це відбувається для всіх тестових зображень при горизонтальному перекритті нижній частині об'єкта. Даний факт пояснюється тим, що при такому перекритті перестають бути видимими істотні частини транспортних засобів (решітка радіатора з фарами, колеса). Як наслідок, при збільшенні відсотка перекриття знижується достовірність присутності об'єкта, а згодом алгоритм вважає, що об'єкта зовсім немає на зображенні. Таким чином, при оцінюванні якості пошуку об'єктів класу «АВТОМОБІЛЬ» з використанням Latent SVM слід

враховувати об’єкти розмітки, які мабуть не менше ніж на 50%. В іншому випадку, необхідно проводити аналіз складу помилки пошуку, що і робиться в даній роботі.

Зауважимо, що для деяких методів і алгоритмів детектування вказаний поріг може бути знижений через їх особливостей. Так метод детектування з урахуванням результатів зіставлення передбачає відновлення положень об’єктів в моменти входу / виходу в кадр, а також дозволяє визначати положення в разі часткових перекриттів. Експериментально даний факт буде підтверджений при аналізі складу помилки пошуку.

За аналогією можна провести аналіз мінімального відсотка видимості транспортних засобів для інших моделей транспортних засобів та алгоритмів пошуку. Дану схему аналізу можна поширити на випадок пошуку довільних об’єктів.

Всі експерименти проводилися з використанням тестової інфраструктури, параметри якої приведені нижче (табл. 3.1).

Таблиця 3.1 – Тестова інфраструктура

Процесор	Intel® Core™ i5 CPU, 3 GHz (2 фізичнх + 2 віртуальних ядра)
Пам’ять	8 ГБ
Операційна система	Windows 7 x64
Середовище розробки	Microsoft® Visual Studio 2010
Бібліотеки	– Intel® Threading Building Blocks 3.0 for Windows, Update 3 – OpenCV 2.4.2

Вибір параметрів для застосування розроблених методів детектування. Параметри алгоритму пошуку Latent SVM. Параметри алгоритму пошуку транспортних засобів є одними з найбільш важливих, тому що він забезпечує коректність виявлення об’єктів. Необхідно вибрати значення двох параметрів:

– параметр моделі класу «АВТОМОБІЛЬ», що визначає мінімальне значення достовірності присутності об’єкта. Слід змінити, оскільки при навчанні моделі використовувалися дані, відмінні від тих, на яких виконується пошук;

– параметр алгоритму пошуку Latent SVM [21]. Пояснимо сенс даного параметра. Алгоритм пошуку для одного і того ж об’єкта може побудувати кілька окаймляючих прямокутників з різними значеннями оціночної функції. Щоб прибрати подібні дублювання використовується «жадібний» алгоритм (non-maximum suppression algorithm), який обчислює коефіцієнт перекриття окаймляючих прямокутників і видаляє положення з меншим значенням оціночної функції і коефіцієнтом перекриття, що перевищує значення зазначеного параметра.

Перший параметр можна підібрати, аналізуючи мінімальний відсоток видимості, при якому спрацьовує алгоритм. Об’єкти на даному відео дрібніше в порівнянні з тестовим відео, тому цілком очікувано, що мінімальний відсоток видимості перевищить 50% [22]. За аналогією виберемо зображення транспортного засобу з пропонованого відео. Побудуємо таблицю спрацьовувань алгоритму пошуку при різних варіантах перекриття і проаналізуємо мінімальний відсоток видимості і значення достовірності [23]. З отриманих результатів нескладно бачити, що алгоритм перестає спрацьовувати на всіх типах перекриттів при 70% видимості. При цьому 3 типу перекриття, на яких об’єкт не виявляється, відносяться до числа найпоширеніших (3 з 7, що становить 43%). Тому для даного відео мінімальний відсоток видимості становить 80%. Згідно значенням достовірності, зазначеним в рядку, що відповідає 20% перекриття, можна зробити висновок, що мінімальне значення достовірності (параметр моделі класу) можна підвищити до 0,245.

Для підбору другого параметра досить запустити алгоритм пошуку Latent SVM з використанням моделі з модифікованим значенням достовірності на одному з кадрів відео і вибрати значення, при якому виходить

найкращий результат (рис. 3.12). Для розглянутого відео параметр алгоритму Latent SVM був обраний рівним 0,02.

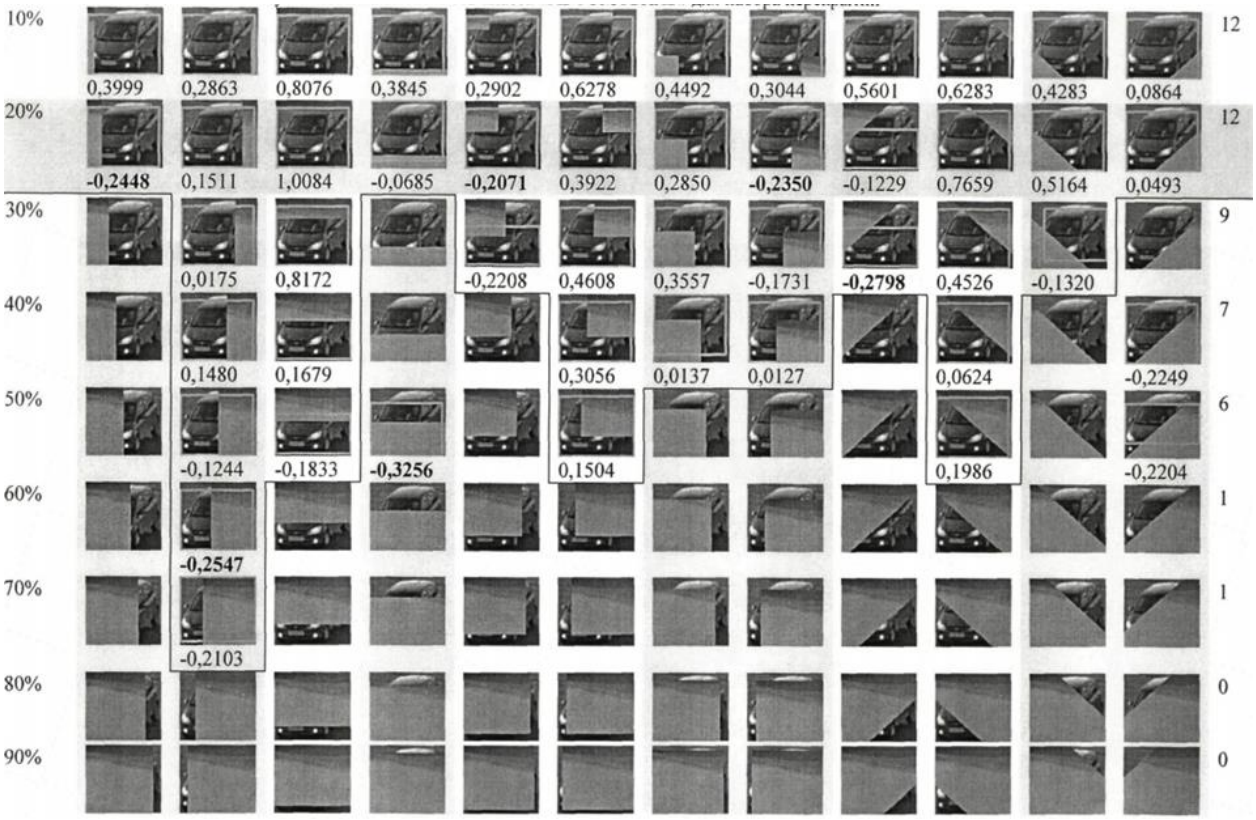

Рисунок 3.12 – Результати пошуку об’єктів класу «АВТОМОБІЛЬ» для набору перекриттів

Обговоримо вибір інших параметрів:

– параметри відсікання по областям інтересу. Очевидно, що кадри відео не містять великої кількості фонових об’єктів. Тому операція відсікання по областям інтересу не потрібно;

– максимальна зміна напрямку руху. На відео спостерігається помітна зміна ракурсу автомобілів – від моменту входу в кадр до моменту виходу різниця між напрямками може доходити до 30°. Найгірший випадок з точки зору методу детектування, коли об’єкт входить в кадр, перекривається іншими об’єктами і знову стає видимим при виході з кадру. При зіставленні видимих положень з урахуванням напрямку руху необхідно зажадати, щоб кут між

спрямування не перевищував 30°. Тому в якості параметра максимального зміни напрямку руху вибирається вказане значення [24];

– мінімальна довжина траєкторії. Якщо зіставити реальні величини об’єктів на тестовому і реальному відео, то нескладно помітити, що розміри відрізняються приблизно в 1,5 рази. Відзначимо, що ракурс автомобілів на відео близький до дзеркального. Тим паче, кут між вертикаллю і напрямом руху трохи різниться. Звідси можна зробити висновок, що камера захоплює ділянку дороги, приблизно вдвічі більший, ніж на тестовому відео. Тому будемо вимагати, щоб довжина траєкторії транспортного засобу була не меншою 20, тобто збільшуємо значення відповідного параметра, використаного при запуску методу на тестовому відео, в 2 рази (рис 3.13).

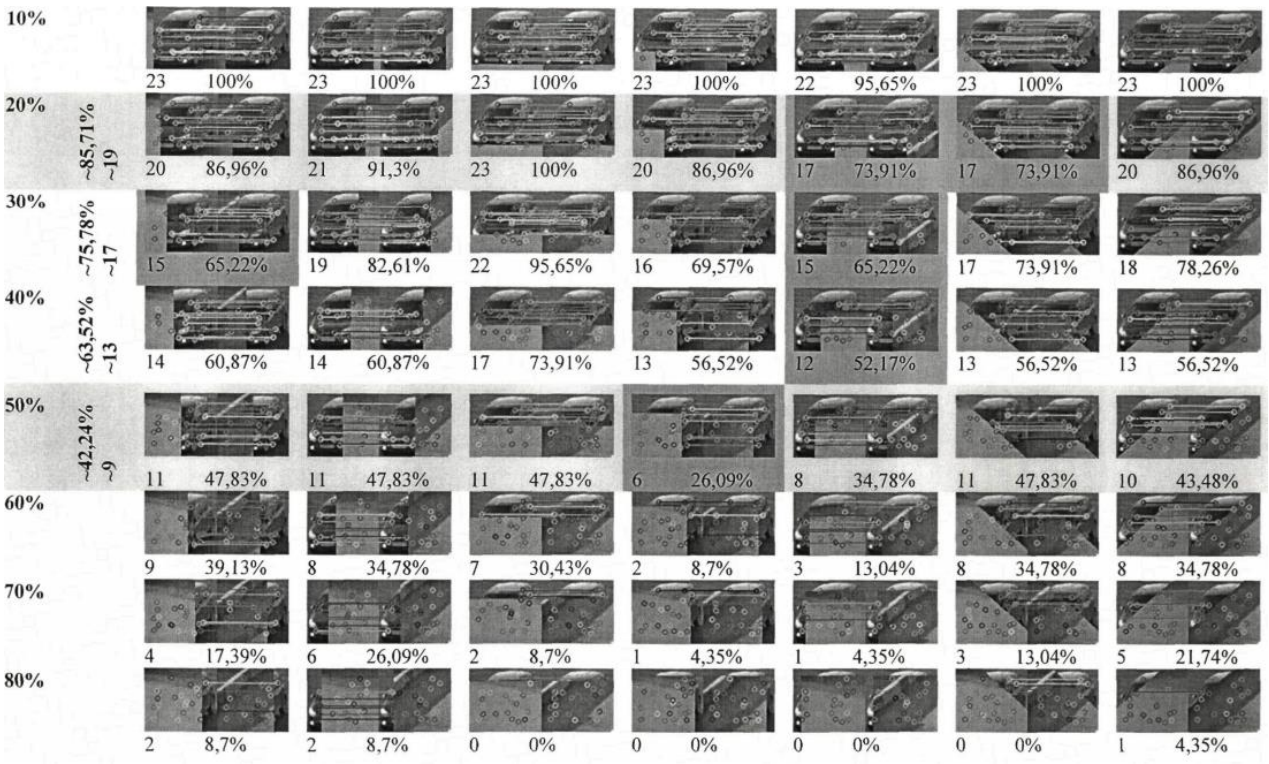

Рисунок 3.13 – Результати зіставлення для пари зображень

3.4 Результати дослідження методів аналізу схеми руху з використанням відеодетектування транспортних засобів

Розглянемо задачу детектування стосовно конкретних вхідних даних (рис 3.14).

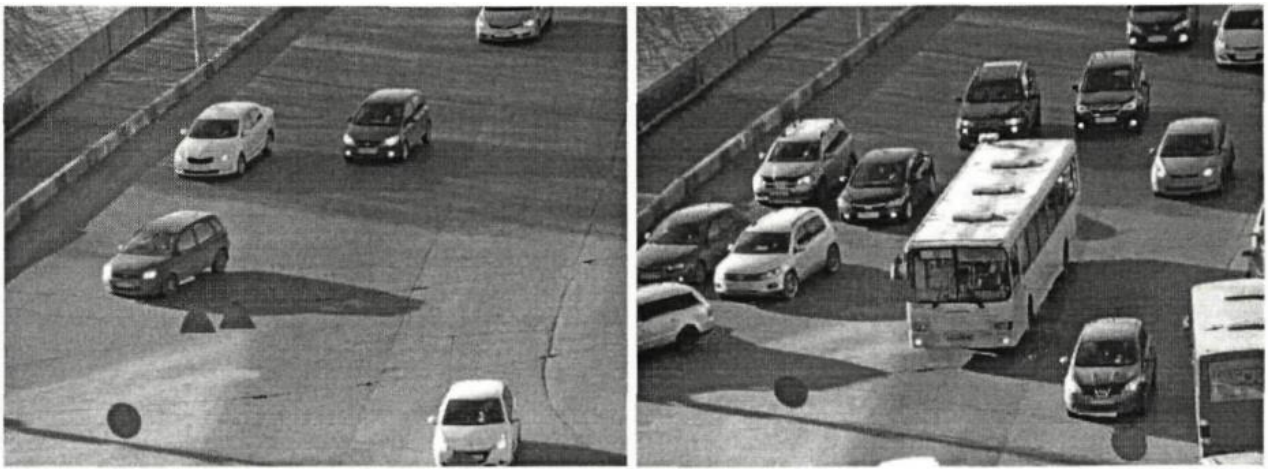

Рисунок 3.14 – Приклади кадрів з відео

Основні характеристики відео:

– дозвіл: 640×480;

– частота: 25 FPS;

– тривалість: 2 хв. 18 с (3456 кадрів);

– загальна кількість окаймляючих прямокутників: 20, 292;

– загальна кількість траєкторій: 117;

– класи транспортних засобів: «АВТОМОБІЛЬ», «АВТОБУС».

У порівнянні з тестовим даними відео володіє наступними відмітними особливостями:

– ракурс транспортних засобів практично дзеркально протилежний ракурсу об’єктів на тестових відео;

– помітна зміна ракурсу об’єктів в процесі руху (від виду спереду до виду в три чверті);

– різниця освітлення у верхній і нижній частині кадру;

– наявність тіней, що займають значну частину кадру;

– наявність розмитих (розфокусувати) зображень внаслідок того, що камера була встановлена на штативі, а зйомка велася в режимі ручної камери;

– невеликий розмір транспортних засобів (приблизно в 1,5 рази менше, ніж на тестовому відео);

– наявність дорожніх знаків і проводів тролейбусних ліній, які розташовані безпосередньо над смугами руху і в певні моменти перекривають істотні частини об’єктів.

Таким чином, запропоноване відео в цілому відрізняється від тестового за багатьма параметрами.

ВИСНОВОК

Основним практичним результатом роботи є створення методу детектування, що забезпечує покадрову класифікацію і відновлення напрямку руху транспортних засобів (в системі координат, пов’язаної із зображенням) із змінним ракурсом. Розроблено набір програмних бібліотек і додатків для детектування транспортних засобів, що реалізують запропоновані методи і необхідний допоміжний функціонал для розмітки відеоданих в напівавтоматичному режимі і кількісного оцінювання результатів пошуку і супроводу. Програмні засоби готові до практичного використання в підсистемах класифікації транспортних засобів, оцінки інтенсивності транспортних потоків з урахуванням напрямку руху та інших.

В роботі отримані наступні наукові і практичні результати:

– запропоновано метод детектування для відновлення траєкторій руху транспортних засобів в системі координат зображення по безліч положень, побудованих з використанням відомого алгоритму пошуку об’єктів різних класів. Метод на відміну від існуючих дозволяє відновлювати траєкторії транспортних засобів, ракурс яких змінюється в процесі руху;

– при певних умовах об’єднувати ділянки траєкторій транспортних коштів, повністю перекриваються на невеликі проміжки часу;

– є можливість оцінювати інтенсивність і якісний склад транспортних потоків різного спрямування, виходячи з траєкторій при наявності інформації про контекст (наприклад, розміщення в’їздів / виїздів на вулиці);

– запропоновано метод детектування, що забезпечує пошук транспортних засобів на деяких кадрах відео потоку і одночасне побудова траєкторій їх руху в системі координат, пов’язаної із зображенням. Метод на відміну від існуючих дозволяє виявляти і супроводжувати транспортні засоби із змінним ракурсом незалежно від класу транспортного засобу;

– є можливість виконувати класифікацію транспортних засобів по одному кадру.

Розроблено архітектуру та реалізована програмна система детектування транспортних засобів, що дозволяє виконувати підготовку тестових даних, розробляти і інтегрувати нові програмні реалізації методів детектування, а також застосовувати розроблені методи детектування і оцінювати якість пошуку і супроводу з точки зору числових показників, розглянутих в роботі. Система може бути використана для детектування об’єктів інших класів, тому що безліч класів визначається набором наявних моделей пошуку.

Виконано експериментальне дослідження ефективності роботи системи та оцінка якості детектування транспортних засобів класу «АВТОМОБІЛЬ» на тестовому і реальному відео.

Результати даного дослідження апробовано на V Міжнародна науково-практична конференція «Study of modern problems of civilization», 19-23 жовтня 2020 р., Осло, Норвегія [25] .

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кучеренко, Є. І., Кучеренко, В. Є., Глушенкова, І. С., & Творошенко, І. С. (2012). Методи, моделі та інформаційні технології оцінювання станів складних об’єктів: монографія.

2. Кучеренко, Е. И., & Творошенко, И. С. (2003). Процессы принятия решений в сложных системах на основе нечетких интервальных представлений. *Вісник Національного технічного університету» ХПІ». Тематичний випуск: Системний аналіз, управління та інформаційні технології.–Х.: НТУ» ХПІ*, *1*(7), 79–86.

3. Кучеренко, Є. І., Творошенко, І. С., & Анопрієнко, Т. В. (2016). Моделювання та оцінювання станів складних об’єктів із застосуванням формальної логіки. *Системи обробки інформації,* (2), 76–82.

4. Кучеренко, Е. И. (2005). Интеллектуальные технологии в задачах принятия решений технологических комплексов на основе нечеткой интервальной логики. *Восточно–Европейский журнал передовых технологий*, (2), 92–96.

5. Мамонов, К. А., & Творошенко, І. С. (2014). Математичні методи і моделі в оцінці нерухомості: навч. посіб.

6. Кучеренко, Е. И., Корниловский, А. В., & Творошенко, И. С. (2010). О методах настройки функций принадлежности в нечетких системах. *Системы управления, навигации и связи*, (1), 13.

7. Tvoroshenko, I. S., & Gorokhovatskyi, V. O. (2020). EFFECTIVE TUNING OF MEMBERSHIP FUNCTION PARAMETERS IN FUZZY SYSTEMS BASED ON MULTI–VALUED INTERVAL LOGIC. Telecommunications and Radio Engineering, 79(2).

8. Tvoroshenko, I. S. (2010). Analysis of Decision–Making Processes in Intelligent Systems, Information Processing Systems, 2.

9. Ahmad, M. A., Tvoroshenko, I., Baker, J. H., & Lyashenko, V. (2019). Modeling the Structure of Intellectual Means of Decision-Making Using a System-Oriented NFO Approach.

10. Ahmad, M. A., Tvoroshenko, I., Baker, J. H., & Lyashenko, V. (2019). Computational Complexity of the Accessory Function Setting Mechanism in Fuzzy Intellectual Systems.

11. Tvoroshenko, I., Ahmad, M. A., Mustafa, S. K., Lyashenko, V., & Alharbi, A. R. (2020). Modification of Models Intensive Development Ontologies by Fuzzy Logic.

12. Matarneh, R., Tvoroshenko, I., & Lyashenko, V. (2019). Improving Fuzzy Network Models For the Analysis of Dynamic Interacting Processes in the State Space. International Journal of Recent Technology and Engineering, 8(4), 1687-1693.

13. Tvoroshenko, I. S., & Gorokhovatskyi, V. O. (2020). EFFECTIVE TUNING OF MEMBERSHIP FUNCTION PARAMETERS IN FUZZY SYSTEMS BASED ON MULTI-VALUED INTERVAL LOGIC. Telecommunications and Radio Engineering, 79(2).

14. Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). IMAGE CLASSIFICATION METHOD MODIFICATION BASED ON MODEL OF LOGIC PROCESSING OF BIT DESCRIPTION WEIGHTS VECTOR. Telecommunications and Radio Engineering, 79(1).

15. Daradkeh, Y. I., & Tvoroshenko, I. (2020). Technologies for Making Reliable Decisions on a Variety of Effective Factors using Fuzzy Logic. Int. J. Adv. Comput. Sci. Appl, 11, 43-50.

16. Gorokhovatskyi, V., & Tvoroshenko, I. (2020). Image Classification Based on the Kohonen Network and the Data Space Modification.

17. Daradkeh, Y.I.; Tvoroshenko, I. Application of an Improved Formal Model of the Hybrid Development of Ontologies in Complex Information Systems. Appl. Sci. 2020, 10, 6777.

18. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). THE APPLICATION OF NON-PARAMETRIC STATISTICS METHODS IN IMAGE CLASSIFIERS BASED ON STRUCTURAL DESCRIPTION COMPONENTS. Telecommunications and Radio Engineering, 79(10).

19. Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Vlasenko, N. V. (2020). USING FUZZY CLUSTERING IN STRUCTURAL METHODS OF IMAGE CLASSIFICATION. Telecommunications and Radio Engineering, 79(9).

20. Szeliski, R., (2010). Computer Vision: Algorithms and Applications, London, Great Britain: Springer-Verlag, 957 p.

21. Abdulkareem, I., Ammar, A., Shubber, Sabah, A., (2018). Multi-criteria decision making to select the best monorail route, Global Journal of Engineering Science and Research Management, pp. 16-32.

22. Peters, J.F., (2017) Foundations of computer vision: Computational Geometry, Visual Image Structures and Object Shape Detection, Cham, Switzerland: Springer International Publisher, 417 p.

23. Sonka M., Hlavac, V., Boyle, R,. (2014) Image Processing, Analysis, and Machine Vision, Atlanta, USA: Thomson-Engineering, 920 p.

24. Duda R.O., Hart P.E., & Stork, D.G. (2000). Pattern classification, Hoboken, USA: John Wiley & Sons, 738 p.

25. Tvoroshenko I., and Dziubenko M. (2020) Modern methods of analysis of the movement scheme using video detection of vehicles, *Abstracts of V International Scientific and Practical Conference «Study of modern problems of civilization» (October 19-23, 2020). Oslo, Norway*, pp. 422-428. DOI: 10.46299/ISG.2020.II.V