МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

Кваліфікаційна робота магістра

«Інформаційна система розпізнавання графічних образів з використанням нейронної мережі»

Здобувач освіти гр. Ін.м-01н Павло ПІДГОРНИЙ

Науковий керівник, Тетяна ЛАВРИК
ст. викладач, к.п.н., доц.

Завідувач кафедри Анатолій ДОВБИШ
д.т.н., проф.

Суми 2022

Сумський державний університет
(назва вузу)

Факультет *ЕлІТ* Кафедра *Комп'ютерних наук*
Спеціальність *«122 - Комп'ютерні науки»*

Затверджую:
зав.кафедрою _____
_____
"_____" _____ 20___р.

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТОВІ

Підгорному Павлу Володимировичу

1. Тема проекту (роботи) *Інформаційна система розпізнавання графічних образів з використанням нейронної мережі*
затверджую наказом по інституту від "____" _____ 20 ___ р. № _____
2. Термін здачі студентом закінченого проекту (роботи)_____
3. Вхідні данні до проекту (роботи) онлайн ресурс github.com, бібліотеки YOLO, зображення для тестування.
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) 1) Аналіз програмного забезпечення для розпізнавання об'єктів (текстів, номерних автомобільних знаків, осіб на відео); 2) Огляд існуючих алгоритмів розпізнавання об'єктів; 3) Проектування інформаційної системи із урахуванням потреб щодо точності розпізнавання та необхідних можливостей; 4) Вибір мови програмування для реалізації програмного продукту; 5) Вибір засобів розробки; 6) Програмна реалізація інформаційної системи; 7) Тестування інформаційної системи.

5. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

6. Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Літературний огляд за досліджуваною проблематикою	Березень 2022 р.	
2	Опис нейромережевих методів, що ґрунтуються на застосуванні штучних нейронних мереж різних типів	Березень 2022 р.	
3	Дослідження методів реалізації систем розпізнавання образів (структурний, статистичний, опорних векторів)	Квітень 2022 р.	
4	Проектування структури системи, вибір мови програмування та середовища розробки	Квітень 2022 р.	
5	Розробка інформаційної системи розпізнавання образів із графічних зображень (тварини, люди, автомобілі)	Травень 2022 р.	
6	Оформлення роботи	12.05.2022 р.	

Студент – дипломник _____
(підпис)

Керівник проекту _____
(підпис)

РЕФЕРАТ

Записка: 105 стор., 38 рис., 5 додатків, 39 літературних джерел.

Об’єкт дослідження – інформаційна система розпізнавання графічних об’єктів з використанням нейронної мережі.

Предмет дослідження – архітектури нейронних мереж, що застосовуються для розпізнавання та класифікації об’єктів на зображеннях.

Мета роботи – розробка інформаційної системи розпізнавання графічних образів з використанням нейронної мережі.

Методи дослідження – інформаційний аналіз, метод моделювання, методи розробки, що базуються на мовах програмування C#, C++, Java.

Результати – проведено аналітичний огляд алгоритмів розпізнавання об’єктів; розглянуто принципи побудови нейронних мереж; модифіковано алгоритм розпізнавання образів; реалізовано алгоритм розпізнавання об’єктів на зображенні; здійснено оцінку ефективності розробленого алгоритму в порівнянні з сучасними альтернативними методами розпізнавання.

Ключові слова: аналіз інформації, розпізнавання об’єктів на зображеннях, бази даних, нейронні мережі, інтелектуальні системи, MySQL, YOLO, phpMyAdmin.

ЗМІСТ

ВСТУП 6
РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ 8
1.1. Аналіз предметної області 8
1.2. Огляд існуючих програмних рішень 13
1.3. Постановка задачі дослідження 21
РОЗДІЛ 2. СИСТЕМА КОНТРОЛЮ ТА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ІЗ ЗОБРАЖЕНЬ 24
2.1. Побудова математичної моделі системи розпізнавання 24
2.2. Аналітичний огляд сучасних методів розпізнавання зображень 30
2.3 Проектування системи розпізнавання зображень 40
2.4. Проектування структури системи, вибір мови програмування та середовища розробки 47
2.5. Архітектурні особливості інформаційної системи 51
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ 57
3.1. Аналіз вимог до системи розпізнавання зображень. 57
3.2. Вибір засобів розробки системи розпізнавання зображень 60
3.3 Бібліотеки та простори імен для реалізації функціоналу 61
3.4 Програмна реалізація та інструкція користувача 64
3.5. Тестування розробленої системи 77
ВИСНОВКИ 80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 81
ДОДАТКИ 85

ВСТУП

Актуальність теми. На сьогоднішній день нейронні мережі виступають переконливою перспективною сферою для дослідження. Вони активно використовуються для вирішення багатьох задач: апроксимація, розпізнавання образів, класифікація та кластеризація, архівація даних, їх стиснення тощо. При коректній реалізації нейронна мережа спроможна ідентифікувати складні взаємозалежності між вхідними інформаційними даними та наявними інформаційними ресурсами та проводити узагальнення.

Саме тому доцільно розглядати нейронні мережі як один із методів моделювання штучних інтелектуальних систем. Процес розпізнавання візуальних образів виступає одним із ключових елементів складних багатокомпонентних систем управління та обробки інформаційних даних, автоматизованих систем тощо. Алгоритмічна обробка і диференціація зображень активно використовується в системах управління безпекою, в системах відеоспостереження, контролю і управління доступом, в інформаційних пошукових системах, системах віртуальної реальності тощо.

Актуальність кваліфікаційної магістерської роботи полягає у дослідженні доцільності використання нейронних мереж для вирішення формалізованих задач в інтелектуальному аналізі даних.

У кваліфікаційній магістерській роботі розглядається та реалізується застосування нейронної мережі при вирішенні проблеми розпізнавання.

Мета кваліфікаційної магістерської роботи – розробка інформаційної системи розпізнавання графічних образів з використанням нейронної мережі.

Поставлена мета зумовила виконання наступних завдань:

1) провести аналіз науково-методичних джерел та існуючих рішень щодо розпізнавання об’єкта на зображенні;

2) дослідити методичні положення процесу розпізнавання на основі нейронних мереж;

3) розглянути принципи побудови нейронних мереж;

4) створити алгоритм розпізнавання образів;

5) реалізувати та описати алгоритм розпізнавання об’єктів на зображенні;

6) оцінити ефективність розробленого алгоритму в порівнянні з сучасними альтернативними методами розпізнавання.

Об’єкт дослідження – інформаційна система розпізнавання графічних об’єктів з використанням нейронної мережі.

Предмет дослідження – архітектури нейронних мереж, що застосовуються для розпізнавання та класифікації об’єктів на зображеннях.

РОЗДІЛ 1
ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз предметної області

На сьогоднішній день теорія розпізнавання образів існує як розділ інформатики та суміжних дисциплін, що розвиває методи класифікації та ідентифікації об’єктів різної природи: сигналів, ситуацій, предметів, що характеризуються вичерпною кількістю деяких ознак . Проблема розпізнавання об’єктів також виділена в розділ міждисциплінарних досліджень – в тому числі включаючи роботу зі створення штучного інтелекту. Також дуже часто використовується при вирішенні практичних завдань у галузі комп’ютерного зору.

Досить розповсюдженим є хибне зіставлення термінів “розпізнавання” та “класифікація”, коли вони розглядаються як синоніми. Але ці терміни не є повністю взаємо замінюваними. Кожен з цих двох термінів може мати свої сфери застосування в залежності від поставленого завдання. Розглянемо загальні елементи моделі класифікації.

Клас – множина об’єктів, що мають спільні властивості. Для об’єктів одного класу передбачається наявність “схожості”. Для задачі розпізнавання може бути визначено довільну кількість класів, більше одного. Кожен клас має свою ідентифікуючу мітку класу. Класифікація – процес призначення міток класу об’єктам відповідно до певного опису властивостей цих об’єктів. Класифікатор – це інструмент для присвоєння міток класам, який в якості вхідних даних отримує перелік ознак об’єкта. До одного з найпоширеніших способів класифікації можна віднести спосіб, що базується на описі об’єктів з використанням ознак, де кожен об’єкт характеризується набором числових або інших специфічних ознак.

Проте існують типи даних, для яких відкриті ознаки не дають високої точності класифікації, наприклад, колір точок зображень або цифровий звуковий сигнал. Загальна класифікація зображень, для прикладу, собак і автомобілів, є

дуже простою для людини і водночас складною для машини. Причиною цього є можливість людини сприймати “приховані ознаки”, недоступні для автоматизованої системи, такі як морда собаки або колеса автомобіля.

Верифікація – процес зіставлення досліджуваного об’єкта із однією моделлю об’єкта або описом класу. Під образом будемо розуміти найменування області в просторі ознак, в якій відображається безліч об’єктів або явищ матеріального світу. Ознакою можна назвати опис тієї чи іншої властивості, що має пряме відношення до предмета або явища. Іншими словами, розпізнавання образів можна визначити як віднесення вихідних даних до певного класу за допомогою виділення істотних ознак або властивостей, які характеризують ці дані, із загальної маси несуттєвих деталей.

Прикладами завдань класифікації є розпізнавання символів, встановлення медичного діагнозу, прогноз погоди, розпізнавання осіб, класифікація документів тощо. Найчастіше вихідним матеріалом служить отримане із камери зображення. Іншим завданням розпізнавання є виділення із вихідних зображень характерних ознак та/або властивостей. Це завдання можна віднести до попередньої обробки. Ознака повинна являти собою характерну властивість конкретного класу, при цьому загальну для цього класу.

При постановці класичної задачі розпізнавання образів прийнято використовувати математичну мову, ґрунтуючись на логічних міркуваннях і математичних доказах. В протилежність до цього підходу існують методи розпізнавання образів з використанням машинного навчання і штучних нейронних мереж, сформовані не такими суворо формалізованими підходами до розпізнавання, але демонструють не гірший, а в деяких випадках значно кращий результат, ніж надають класичні методи.

Характер проблематики розпізнавання образів також обмежує сферу застосування різних традиційних методів вузькими спеціалізованими напрямками, в кожному з яких найбільш ефективними виявляються одні методи і неефективними інші. Цей фактор обумовлює складність загального вирішення проблеми індукції й в цій дисципліні.

Розпізнавання образів – завдання ідентифікації об’єкта або визначення будь-яких його властивостей за його зображенням (оптичне розпізнавання) або аудіозаписом (акустичне розпізнавання). Одним із базових понять, що немає конкретного формулювання є поняття множини. У комп’ютері множина представляється набором неповторюваних однотипних елементів. Слово “неповторювані” означає, що якийсь елемент присутній в множині або відсутній. Універсальна множина включає всі можливі для розв’язуваної задачі елементи, порожня множина не містить жодного елемента.

Образ – класифікаційне угруповання в системі класифікації, яка об’єднує певну групу об’єктів за певною ознакою. Образи наділені характерними об’єктивними ознакам, які полягають у тому, що різні суб’єкти сприйняття навчаються на різних вхідних даних, переважно незалежно та однаково один від одного диференціюють одні й ті ж об’єкти. У класичній постановці задачі розпізнавання універсальна множина розбивається на частини-образи. Кожне відображення на сприймаючі органи системи будь-якого об’єкта, не враховуючи положення цього об’єкта щодо таких органів прийнято ідентифікувати зображенням об’єкта, а множини цих зображень, які пов’язані характерними рисами, виступають образами.

Вважаємо доцільним визначити головні відомі підходи до питання розпізнавання образів. Як перший підхід можна зазначити метод перебору. У цьому випадку проводиться порівняння з базою даних, де для кожного виду об’єктів представлені різноманітні модифікації відображення. Наприклад, для оптичного розпізнавання образів можна застосувати метод перебору виду об’єкта під різними кутами, масштабами, зміщеннями, деформаціями. Для літер потрібно перебирати шрифт, властивості шрифту.

Другий підхід – знайти контур об’єкта й досліджувати його властивості (зв’язність, наявність кутів). Існує безліч алгоритмів для виділення контурів, такі як: оператор Лапласа, оператор Собеля, оператор Кенні тощо. Детектування контурів необхідно в тому випадку, коли є досить складне зображення і використовуючи малі інструменти бібліотеки комп’ютерного зору необхідно виділити об’єкти на цьому зображенні для подальшої роботи з ними.

Одним із важливих підходів є використання штучних нейронних мереж (багатошарові персептрони, карти Кохонена). Цей метод вимагає спеціальної структури нейронної мережі, яка буде враховувати особливості даної задачі або потребувати великої кількості прикладів із правильними відповідями (прикладів задач розпізнавання).

Також існує експертний метод, заснований на безперервному навчанні експертної системи в процесі експлуатації.

Ключові задачі процесу розпізнавання образів:

– ідентифікація тексту;
– ідентифікація автомобільних номерів;
– ідентифікація об’єктів на картах;
– ідентифікація обличчя та інших біометричних даних;
– ідентифікація штрих-кодів;
– ідентифікація мови.

Розпізнавання образів є перспективним напрямком у рекламі та маркетингу. Нейронні мережі дозволяють за лічені години дізнатися речі, для пошуку яких в інших випадках потрібна велика команда професіоналів і тижні, а то й місяці досліджень. Наприклад, сервіс *YouScan*, система моніторингу соціальних медіа, відстежує згадки брендів в соціальних мережах. Причому робить це не тільки в тексті, але й на фотографіях, а також допомагає зробити певні висновки про продукт. За допомогою розпізнавання образів на фото знаходять певні закономірності, що можуть стати в нагоді для таргетування реклами.

Розпізнавання образів на камерах міського відеоспостереження – найперспективніше використання машинного зору. Такі системи розумного відеоспостереження можуть бути використані з метою ідентифікації злочинців в місцях масового скупчення людей, керування доступу до закритих територій, а також використовувати зібрану інформацію для різних досліджень.

Розпізнавання образів вже стало справжнім проривом в медицині. У багатьох випадках комп’ютери помічають речі, які пропускають навіть найдосвідченіші лікарі. Створена *Google* нейромережа *Inception* аналізує

мікроскопічне дослідження біопсії лімфатичних вузлів в пошуку ракових клітин в молочних залозах. Для людини це дуже довгий і трудомісткий процес, в процесі якого легко помилитися або пропустити щось важливе, оскільки в деяких випадках розмір зображення становить 100 000 х 100 000 пікселів.

Розпізнавання об’єктів в автомобілях – це необхідна частина систем безпеки *ADAS* (*Advanced driver-assistance systems*), які можуть бути реалізовані як складними засобами, на зразок інфрачервоних датчиків, так і за допомогою монокулярної камери.

У дронах процес ідентифікації використовуються в цілях розваги, так і з науковою метою. Дрони з функцією розпізнавання використовується і для виконання серйозніших завдань. Для прикладу, компанія *eSmart Systems* створила інтелектуальні рішення для енергомереж. В межах одного з їхніх проєктів дрони використовуються для пошуку несправностей на лініях електропередач. Також дрони можуть бути застосовані для військових специфічних цілей – для ідентифікації та подальшого супроду об’єктів війського призначення.

Не менш важливим на сьогоднішній день виступає питання ідентифікації людських обличь з метою захисту інформації, особливо персональних даних. Автоматична ідентифікація осіб виступає прикладом такої технології, яка в свою чергу, аналізує ключові характерні риси особи, розроблює його математичну репрезентацію, а потім, з метою визначення потенційних збігів, порівнює з базою даних відомих фізичних осіб.

Процес ідентифікації образів – це ціль розпізнавання об’єкта або виокремлення характерних йому ознак за його зображенням або аудіозаписом. Досить проблемним технічним питанням виступає розробка штучних систем, наділених функцією ідентифікації образів.

Взагалі питання розпізнавання образів складають дві комплексні частини: розпізнавання та навчання. Навчання реалізується методом показу конкретних об’єктів із окресленням їх належності певного способу. Результатом навчання виступає функція системи реагування однаковими типовими реакціями на весь перелік об’єктів одного конкретного образу, а іншим типом реакції на об’єкти

несхожих образів. Важливим є той факт, що під час навчання окреслюються не лише об’єкти та їх належність конкретному образу, а й процес ідентифікації нових образів, що описує діяння уже навченої системи. Проблему навчання становить процес автоматизації цієї процедури.

Досить широким є коло завдань, які розв’язуються за допомогою розпізнавальних систем. Такий перелік складають завдання ідентифікації зорових та слухових образів, задача диференціації складних явищ та процесів тощо. Перед стартом аналізу конкретного об’єкта варто одержати належні впорядковані інформаційні дані про нього.

Однією із ключових проблемних питань процесу розпізнавання виступає вибір вихідного опису. У випадку вдалого вибору такого об’єкту, що являє собою простір ознак, задачі розпізнавання можуть стати тривіальними і, навпаки, до досить проблемної подальшої трансформації інформації або відсутності рішення може призвести некоректно обране вихідне рішення.

1.2. Огляд існуючих програмних рішень

На даний момент основний напрямок розвитку нових інформаційних технологій, що впроваджуються в життя людини, – це поліпшення людино-машинної взаємодії. Варто відзначити, що кількість заходів, присвячених цьому напрямку, досить значна. Даний напрямок є дуже перспективним, про що говорять щорічні фінансові вливання та збільшення щорічно числа лабораторій, що займаються розробкою систем машинного зору.

З кожним роком зростання інтересу до технологій розпізнавання об’єктів помітно збільшується. На сьогоднішній день вже існує кілька десятків систем розпізнавання, що застосовуються у різних сферах людського життя. Всі вони є особливими, з власними перевагами та недоліками, є програми, направлені на розпізнавання обличь, інші – на розпізнавання емоцій чи тексту, деякі програми є досить вузькопрофільними – спрямовані на розпізнавання порід тварин чи

навіть монети. Розглянемо найпопулярніші існуючі програмні рішення у сфері розпізнавання зображень:

– *CuneiForm* (рис. 1.1) – система оптичного розпізнавання символів, розроблений російською компанією *Cognitive Technologies*. Програма перетворює файли зображень, отримані зі сканера або іншим шляхом на текст. Алгоритми, закладені в *CuneiForm,* ґрунтуються на правилах написання букв, на їхній топології, і не вимагають задавання певних еталонів або навчання. Розпізнаються будь-які друкарські шрифти – книги, газети, журнали, роздруківки з лазерних і матричних принтерів, тексти з друкарських машинок тощо.

Переваги програмного рішення:

1) розпізнавання тексту 20 найпопулярнішими мовами світу (українська, англійська, російська, німецька, польська, іспанська та інші);

2) підтримка різних друкованих шрифтів;

3) перевірка за словником розпізнаного тексту;

4) збереження структури документа;

5) алгоритми оптичного розпізнавання (*OCR, Optical Character Recognition),* вбудовані в програму, дозволяють розпізнавати текст з матричного принтера, неякісних ксерокопій та факсів.

Недоліками даного програмного рішення є такі:

1) відсутня підтримка надто великих документів та файлів (більше 400 dpi);

2) не підтримуються деякі типи сканерів (для такого функціоналу вимагається встановлення додаткового програмного забезпечення);

3) досить слабкий дизайн програми.

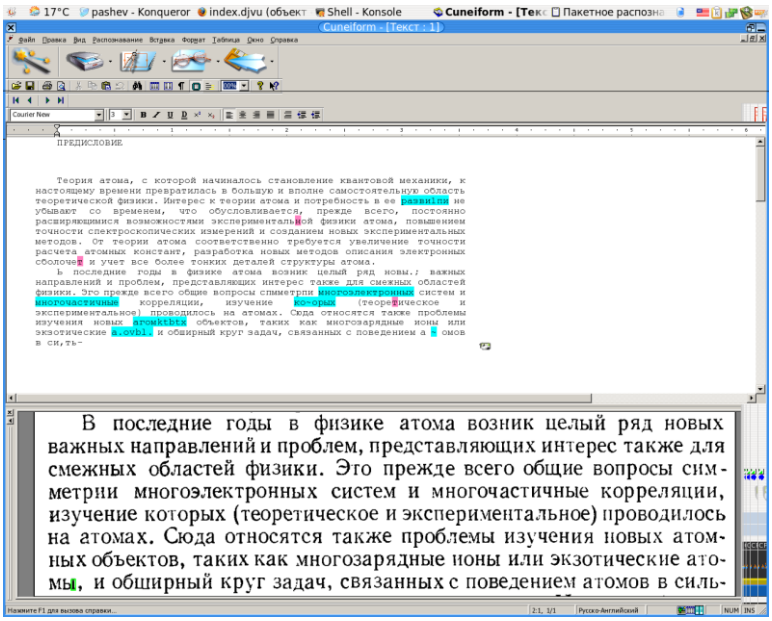

Рисунок 1.1. – *CuneiForm* – програма розпізнавання тексту

– *NumberOk* (рис. 1.2) – платне програмне забезпечення для розпізнавання автомобільних номерних знаків з вбудованим функціоналом контролю і управління доступом автотранспорту, підрахунку кількості автомобілів і розрахунку часу перебування на території. Щодо переваг такого програмного рішення можна виділити наступні:

1) програма сумісна з будь-якою версією *Windows* і працює з усіма камерами, що підтримують передачу *RTSP* потоку;

2) працює з аналоговими *DVR*;

3) великий перелік країн, номерні знаки яких програма може розпізнати (Україна, Білорусь, Молдова, Росія, країни Євросоюзу, Туреччина, Ізраїль).

До недоліків варто віднести наступне:

1) висока вартість програмного рішення (окремо сплачується вартість самого програмного забезпечення – ціна залежить від кількості каналів охоплення та точок доступу);

2) некоректне розпізнавання інформації на високих швидкостях автомобіля.

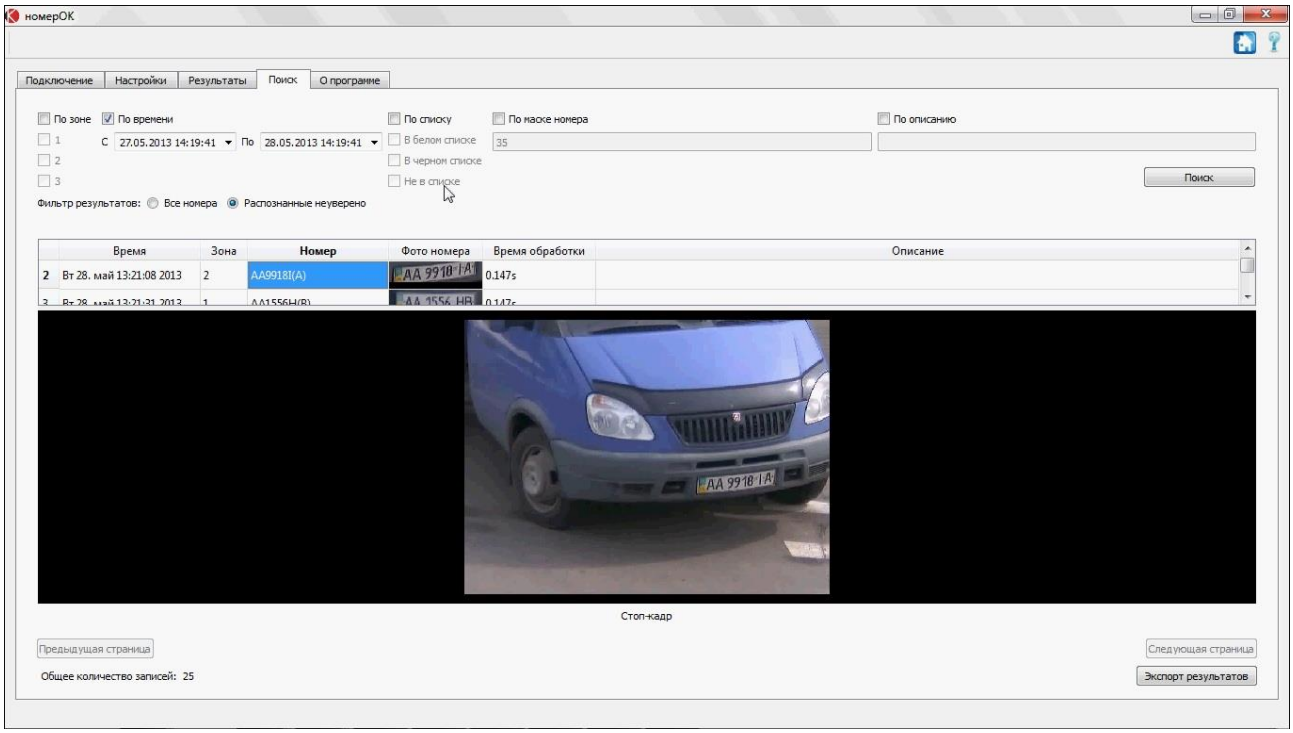

Рисунок 1.2 – *NumberOk* – програма розпізнавання номерних автомобільних знаків на відео

– *FaceVACS-VideoScan* (рис. 1.3) – програмне забезпечення, яке призначене для розпізнавання осіб з відеопотоку в режимі реального часу, досить просте та зручне для використання, розроблене компанією *Cognitec Systems*.

Дане системне рішення складається з кількох програмних компонентів:

- відеосервера, що управляє відеопотоками;
- сервера відеосканування, який координує всі компоненти системи та виконує основні біометричні операції;
- обчислювального вузла, що використовується для розподілу обчислювального навантаження;
- інтерфейсу користувача;
- диспетчера сигналів, що отримує повідомлення про події та обслуговуючі мобільні пристрої;
- операційної бази даних;
- комплекту інтеграторів.

На сьогоднішній день технологія *FaceVACS* використовує алгоритм розпізнавання обличь. Цей алгоритм стійкий до змін міміки, поворотам особи (на ±15º), частковому закритті обличчя, використанню окулярів тощо.

Крім того, програмне забезпечення *FaceVACS* має такі особливості:

1) стеження в режимі реального часу на кількох відеопотоках;

2) можливість масштабованості;

3) надзвичайно висока обробка порівнянь за зразками (у середньому – 900 000 порівнянь у базі за секунду);

4) інтеграція з веб-камерами, *http*–камерами, цифровими фотоапаратами, відеокамерами, а також підтримка зображень у найпоширеніших форматах;

5) об'ємні бази даних, інтеграція з Oracle, IBM DB2, MSSQL Server.

Щодо недоліків зазначимо наступні:

1) освітлення зображення відіграє вагому роль, адже програма не призначена для розпізнавання зображення з поганим освітленням (зображення обличчя в тіні система не розпізнає);

2) для належної безперебійної роботи програма висуває вимоги до характеристик автоматизованого робочого місця та якості мережі (при низькому з’єднанні програма не буде коректно вирішувати поставлені завдання).

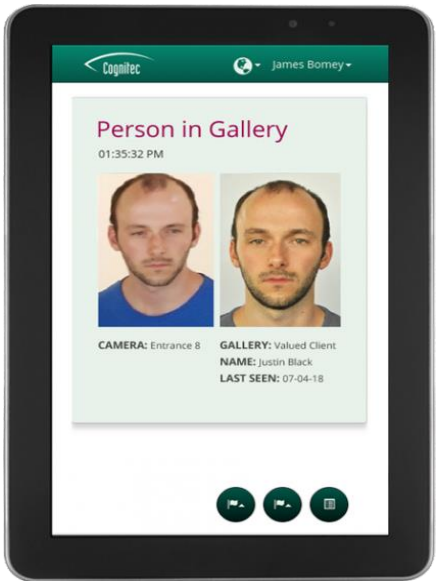

Рисунок 1.3 – *FaceVACS-VideoScan* – програма розпізнавання осіб на відео

– *CarGo Enterprise* (рис. 1.4) – автоматизована система управління контрольно-транспортним пунктом, призначена для організації контролю проїзду автомобільного транспорту через точки контролю з організацією необхідного алгоритму проїзду і збору даних про автомобіль.

До числа даних для контролю входять такі показники:

- державний реєстраційний номер автомобіля (номерний знак);
- напрямок руху (в'їзд/виїзд);
- повноваження автомобіля (приналежність групі доступу);
- дата/час проходження точки контролю.

Підтримуються основні формати однорядкових і дворядкових номерних знаків багатьох країн Європи, Азії, Південної та Північної Америки (як загальногромадянські, так і спеціалізовані – дипломатичні, військові, поліцейські, транзитні тощо), розпізнаються арабські цифри, а також латинські та кириличні символи алфавітів.

Модуль функціонує, в основному, в автоматичному режимі. При появі автомобіля на в'їзді розпізнає державний номер автомобіля і виконує відкриття шлагбауму для в'їзду на територію, якщо це вимагається правилами функціонування.

Програмою допускається одночасне керування до 16 односпрямованими точками доступу або з урахуванням того, що кожна точка доступу може використовуватися в двох напрямках – для в'їзду та виїзду.

Серед переваг програмного забезпечення *CarGo Enterprise* виділимо такі*:*

1) висока точність розпізнавання (до 94%);

2) можливість керування пропускним процесом на базі виконаного розпізнавання (якщо авто, яке проїжджає, відповідає всім необхідним вимогам, система має повноваження відкрити шлагбаум для проїзду);

3) автоматичний режим функціонування.

Недоліками є:

1) висока вартість встановлення програмного забезпечення;

2) програма дуже містка, тому займає значний обсяг пам’яті на комп’ютері;

3) програма вузькоспеціалізована – її використання не передбачене для пересічних користувачів, для її експлуатування необхідні додаткові дозволи.

Рисунок 1.4 – *CarGo Enterprise* – автоматизована система управління контрольно-транспортним пунктом

– *NEC's Face Recognition* (рис. 1.5) – є однією з досить відомих систем розпізнавання осіб, розроблена японською компанією «*NEC*», що дозволяє ідентифікувати людей за кадрами багаторічної давнини і навіть, якщо людина перебуває в окулярах або зображує гримасу. Всі розпізнані особи зберігаються в базі даних, тому у разі потреби можна підняти всю історію відеореєстрації та переглянути дату та час будь-якого збереженого зображення.

NEC аналізує індивідуальні особливості особи (розмір, форму зіниць, лінії носа та рота), їх взаємне розташування, і знаходить потім за цією інформацією відповідну людину в базі даних.

Система розпізнавання обличь *NEC Face* має такі переваги:

1) можливість спостереження та контролю в реальному часі;
2) ідентифікація на основі індивідуальних рис особи;
3) можливість пошуку подій у базі даних;
4) ведення журналу зображень осіб;
5) стійкість до повороту обличчя на ±15º та нахилу голови до 45º у будь-якому напрямі від переднього становища;

До недоліків такого програмного рішення можна віднести наступне:

1) відсутність вітчизняної версії для інсталяції та користування;
2) низька точність розпізнавання осіб, що мають додаткові аксесуари (окуляри, маска, кепка тощо).

Рисунок 1.5 – *NEC's Face Recognition* – система розпізнавання осіб

Варто заначити, що на сьогоднішній день програми розпізнавання знайшли своє відображення і для спрощення життя пересічного користувача. Досить популярними стали телефонні додатки, що працюють з ідентифікацією обличь, наприклад, для щоденного розблокування мобільних телефонів.

– *FaceLock* (рис. 1.6) – це безкоштовний мобільний додаток, який захищає інші програми, встановлені на пристрої, використовуючи лише обличчя користувача. Принцип роботи цього програмного рішення наступний – на початку роботи з додатком користувач у режимі реального часу робить кілька знімків обличчя у різних формах. Програма запам’ятовує дану інформацію. Далі коли користувач захоче увійти до необхідного додатку, програма просканує обличчя користувача, зіставивши із стартовими фото, і у позитивному випадку надасть доступ до програми.

Переваги:

1) низькі системні вимоги;
2) сумісність з усіма операційними платформами;
3) можливість налаштування рівня безпеки.

Недоліки:

1) можливість встановлення захисту лише для одного додатку;
2) висока частота помилок.

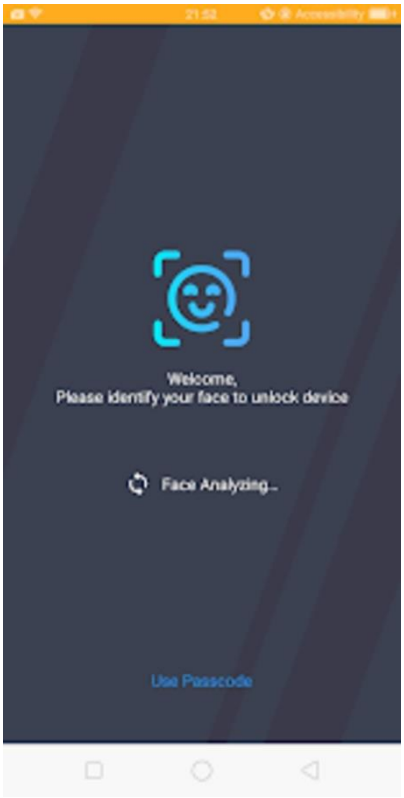

Рисунок 1.6 – мобільний додаток *FaceLock*

1.3. Постановка задачі дослідження

Основною метою кваліфікаційної магістерської роботи є аналіз існуючих нейро-мережевих алгоритмів та методів розпізнання об’єктів на зображеннях у реальному масштабі часу для їх подальшої ідентифікації та використання у різних напрямках. Також отримані дані можна використовувати для збору даних та отримання різних статистичних залежностей. У процесі виконання даної роботи необхідно проаналізувати методи розпізнавання та системи, що їх реалізують. У даній роботі будуть детально розглядатися процес розробки інформаційної системи для обробки та розпізнавання зображень.

Визначимо основні завдання, що необхідно вирішити в кваліфікаційній магістерській роботі:

– аналіз існуючих методів розпізнання;

– дослідження нейронно-мережевих методів для розпізнавання зображень та виявлення переваг та недоліків кожного з нейронно-мережевих методів;

– вибір методу нейронної мережі, що максимально ефективно задовольнить процес вирішення завдання ідентифікації об’єктів на фотокартці у режимі реального часу;

– вибір актуального середовища для розробки інформаційної системи, аналіз його переваг та недоліків;

– вибір і аналіз додаткових технологій, що можуть знадобитися для ефективного процесу розробки;

– реалізація інформаційної системи;

– детальний аналіз реалізованого продукту, проведення тестування, визначення вектору подальшого розвитку.

Отже, підсумувавши наведене вище у розрізі теми кваліфікаційної магістерської роботи необхідно розробити інформаційну систему для ідентифікації об’єктів на зображенні у режимі реального часу, що для належного виконання своїх функцій у роботі буде використовувати найактуальніші та ефективні методи та технології. Також доцільним буде проведення аналізу результатів роботи та швидкодії застосунку, окреслити існуючі переваги та

недоліки, а також зробити висновки щодо доцільності даної системи та зазначити вектор можливого розвитку програмного застосунку.

Для вирішення зазначених завдань передбачається використання методів штучного інтелекту, методів розпізнавання образів, штучних нейронних мереж, моделей і методів нечіткої логіки, методів обробки цифрових зображень, еволюційне моделювання.

РОЗДІЛ 2
СИСТЕМА КОНТРОЛЮ ТА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ІЗ ЗОБРАЖЕНЬ

2.1. Побудова математичної моделі системи розпізнавання

В останні десятиліття в світі бурхливо розвивається нова прикладна область математики, що спеціалізується на штучних нейронних мережах. Штучні нейронні мережі – математичні моделі, а також їх програмні або апаратні реалізації, побудовані за принципом організації та функціонування мереж нервових клітин живого організму. Актуальність досліджень в цьому напрямку підтверджується чисельними прикладами застосувань нейронних мереж. Це автоматизація процесів розпізнавання образів, адаптивне управління, апроксимація функціоналів, прогнозування, створення експертних систем, організація асоціативної пам'яті та багато інших додатків.

За допомогою нейронних мереж можна, наприклад, передбачати показники біржового ринку, виконувати розпізнавання оптичних або звукових сигналів, створювати самонавчальні системи, здатні керувати автомашиною при парковці або синтезувати мову за текстом.

Широке коло завдань, що вирішується нейронними мережами, не дозволяє в даний час створювати універсальні, потужні мережі, змушуючи розробляти спеціалізовані мережі, що функціонують за різними алгоритмами. Проте, тенденції розвитку нейронних мереж ростуть з кожним роком. Одним з перших завдань, що вирішуються за допомогою нейронних мереж, було розпізнавання образів на графічних зображеннях. З того часу було запропоновано досить багато абсолютно нових рішень, були вдосконалені багато відомих рішення і алгоритми [12].

Вважаємо доцільним підсумувати позитивні риси розподіленої обробки інформації в штучних нейромережах:

– Паралелізм обробки інформації – глобальність зв’язків між нейронами. До навчання ці зв’язки довільні, але навчання на прикладах “проявляє” конкретну структуру мережі під конкретне завдання.

– Висока швидкість. Вона можлива внаслідок внутрішнього паралелізму мережі та простої реалізації у вигляді швидкодіючих електронних мікросхем.

– Єдиний і ефективний принцип навчання нейромереж – мінімізація емпіричної помилки шляхом її зворотного поширення мережею. Ззовні задається лише мета навчання – тобто спосіб визначення помилки за виходами мережі. Далі мережа поступово модифікує свою конфігурацію, мінімізуючи цю помилку, як наслідок – все краще справляючись з покладеним на неї завданням.

– Надійність функціонування. Надмірність зв’язків призводить до того, що значення кожного із них окремо не відіграє вирішальної ролі. Виведення з ладу обмеженого числа нейронів або обрив деяких зв’язків не позначаються критичним чином на якості роботи всієї мережі.

– Здатність вирішувати неформалізовані завдання – випливає зі здібності нейромереж самостійно виробляти досить складні алгоритми обробки даних, формалізувати які самостійно часто не можуть навіть найкращі експерти даної предметної галузі. Звідси – відносна дешевизна нейромережевих розробок.

– Перепрограмування. Нейронні мікросхеми легко адаптуються до нових умов експериментів [3].

На практиці нейромережі використовуються в двох іпостасях – як програмні продукти, що виконуються на звичайних комп’ютерах, і як спеціалізовані апаратно-програмні комплекси. У першому випадку не використовується вбудований паралелізм нейромережевих алгоритмів. Для багатьох завдань в додатках *data mining* – при аналізі та узагальненні баз даних, особливо швидкої дії і не потрібно: для них цілком вистачає продуктивності сучасних універсальних процесорів. Для другої групи додатків – обробки сигналів в реальному часі, паралелізм нейро-обчислень є критичним фактором.

Нейромережеві методи, засновані на застосуванні різних типів штучних нейронних мереж , останнім часом набули широкого поширення.

Основні завдання, які вирішуються за допомогою нейронних мереж:

– розбиття простору ознак на області, відповідні класи (класифікація, розпізнавання, кластеризація);

– опис ключових характеристик, стиснення і реконструкція образів;

– апроксимація функції багатьох змінних з будь-якою заданою точністю;

– прогнозування часових рядів;

– асоціативна пам’ять;

– вирішення оптимізаційно-комбінаторних задач;

– розпізнавання з урахуванням топології простору [22].

Більшість з цих завдань прямо або побічно пов’язані з розпізнаванням зображень.

Налаштування нейронної мережі для вирішення певної задачі проводиться в процесі навчання на наборі тренувальних прикладів. Таким чином, не потрібно вручну визначати параметри моделі (вибирати ключові ознаки, враховувати їх взаємовідношення тощо) – нейронна мережа витягує параметри моделі автоматично найкращим чином в процесі навчання. Залишається тільки побудувати тренувальну вибірку. В задачах класифікації при цьому відбувається неявне виділення ключових ознак всередині мережі, визначення значимості ознак і системи взаємовідносин між ними. В даний час розроблені потужні, гнучкі й універсальні механізми навчання різних типів нейронних мереж .

Нейронні мережі мають гарну узагальнюючу здатність. Це означає, що досвід, отриманий в процесі навчання на кінцевому наборі образів, нейронна мережа може успішно поширювати на всю безліч образів. Крім інтерполяційних узагальнюючих здібностей, нейронні мережі можуть добре екстраполювати, тобто застосовувати свій досвід на якісно інші образи, ніж ті, які зустрічалися в узагальнюючої вибірці.

Архітектура нейронної мережі реалізується на паралельних обчислювальних засобах: спеціалізованих мікросхемах, оптичних і квантових комп’ютерах. Це відкриває широкі перспективи застосування нейронних мереж в майбутньому. Нейронна мережа характеризується нечітким і розподіленим

зберіганням інформації, тобто немає окремого нейрона, що відповідає за будь-яке поняття або ознаку, і видалення або спотворення роботи цього нейрона не призведе до фатальних наслідків [19].

За характером функціонування нейронні мережі можуть бути однопрохідними, коли вихід мережі розраховується за один прохід мережі, і релаксаційним, коли функціонування мережі триває до досягнення стабільного стану.

Нейронні мережі найбільш пристосовані до вирішення широкого кола завдань, так або інакше пов'язаних з обробкою образів. Визначимо список типових постановок задач для нейронних мереж:

– Апроксимація функцій з набору точок (регресія).

– Розпізнавання, класифікація даних по заданому набору класів.

– Кластеризація даних з виявленням заздалегідь невідомих класів-прототипів.

– Стиснення інформації.

– Відновлення втрачених даних.

– Асоціативне перетворення інформації.

За цими завданнями проглядається якийсь єдиний прототип, що дозволяє при відомій частці уяви зводити їх один до одного. Найчастіше – це типові приклади некоректних задач, тобто задач, що не мають однозначної відповіді, алгоритми яких невідомі або не мають суворого обґрунтування.

Напевно, в кожній предметній області при найближчому розгляді можна знайти постановки нейромережевих завдань. Список областей, де рішення такого роду завдань має практичне значення вже зараз, дуже широкий:

– Економіка і бізнес: передбачення ринків, автоматичний дилинг, оцінка ризику неповернення кредитів, передбачення банкрутств, оцінка вартості нерухомості, оптимізація портфелів, оптимізація товарних і грошових потоків, автоматичне зчитування чеків і форм, забезпечення безпеки транзакцій за пластиковими картками.

– Політичні технології: аналіз і узагальнення соціологічних опитувань, передбачення динаміки рейтингів, виявлення значущих чинників, об'єктивна кластеризація електорату, візуалізація соціальної динаміки населення.

– Безпека і охоронні системи: системи ідентифікації особистості, розпізнавання голосу, осіб в натовпі, розпізнавання автомобільних номерів, аналіз аерокосмічних знімків, моніторинг інформаційних потоків, виявлення підробок.

– Введення і обробка інформації: обробка рукописних чеків, розпізнавання підписів, відбитків пальців і голоси, уведення в комп'ютер фінансових і податкових документів.

Нейронні мережі – це не що інше, як новий інструмент аналізу даних. І краще інших їм може скористатися саме фахівець у своїй предметній області. Основні труднощі на шляху ще більш широкого поширення нейронних технологій – в невмінні широкого кола професіоналів формулювати свої проблеми в термінах, що допускають просте нейромережеве рішення, тобто невміння проводити нейромережевому декомпозицію розв'язуваної задачі.

Основні класи завдань, що вирішуються за допомогою нейропакетів.

– Розпізнавання (Класифікація).

– Кластеризація.

– Регресія (прогнозування, передбачення).

– Зниження розмірності.

У завданнях регресії метою є оцінка значення числових (приймаючої безперервний діапазон значень) вихідних змінних за значеннями вхідних змінних. У завданні розпізнавання мережа повинна віднести кожне спостереження до одного з декількох класів [18]. Завдання класифікації (кластеризації) полягає в угрупованні схожих образів в класи (кластери).

Використання нейронних мереж для вирішення перерахованих завдань передбачає виконання типової послідовності дій:

– Отримання вихідних даних.

– Відбір вхідних даних і зниження розмірності.

– Оцифрування даних.
– Вибір доцільної архітектури мережі.
– Навчання нейронної мережі.
– Тестування нейронної мережі.
– Отримання готового рішення [29].

На стартовому етапі було обрано об’єкт дослідження, що цілком відповідає поставленим задачам та вимогам кваліфікаційної магістерської роботи. Суть нашої програми досить проста, не обтяжена специфікою предметної області, але, в той же час має значний потенціал трансформації в реальну систему, для прикладу охоронна компанія чи специфічні галузеві служби пошуку тощо.

Розробка нашої програми починається з її проектування. Проектування розроблюваної інформаційної системи має на увазі під собою продумування та планування базового функціоналу, які буде виконувати програма.

Інформаційна система, що розроблюється, має відповідати деяким функціональним вимогам.

Основні функції, які надає система:

– можливість розпізнавання зображення;
– визначення відсоткового співвідношення достовірності розпізнаного об’єкту;
– збереження у системі розпізнаних зображень та об’єктів на них;
– можливість розпізнавання одразу кількох зображень;
– можливість розпізнавання зображень з Інтернет-ресурсів;
– маніпуляції з даними розпізнаних об’єктів (почасова фіксація тощо).

Створена інформаційна система пропонує певну функціональність для роботи із зображеннями, а саме при використанні програми, вона дозволяє користувачеві обрати тип та формат зображення. Користувач може спостерігати весь процес розпізнавання та буде сповіщений для ознайомлення із результатом.

Також реалізована функція попередження користувача у випадку некоректної роботи із програмою – якщо користувач натисне кнопку початку роботи без попередньо обраного зображення, програма його про це попередить. Крім того, користувач може переглянути історію розпізнавання.

2.2. Аналітичний огляд сучасних методів розпізнавання зображень

На даний момент існує значна кількість методів реалізації системи розпізнавання образів, найпоширеніші серед яких розглянемо нижче.

Розпізнавання образів – складний комплексний процес ідентифікації об’єкта, визначення усіх його характерних властивостей за зображенням та зарахування його до площини образів за визначеним правилом.

Існують три основні групи методів розпізнавання образів.

– Структурний (лінгвістичній) підхід застосовується до задач розпізнавання образів, в яких важлива інформація про структуру конкретного об’єкта. Від процедури розпізнавання потрібна здатність у визначенні об’єкту його класу та відомості про об’єкт, який не дозволяє віднести його до іншого класу. Структурні методи ідентифікації об’єктів характеризуються своєю структурою значень і відношень. Структурний підхід ґрунтується на аналогії між синтаксисом та структурою реалізації образу. Головна перевага структурних методів – це можливість створити велику кількість реалізацій представивши невелику множину елементів. Недоліками є обмеженість використання, прямі вирішальні правила не існують в даному методі. Метод ґрунтується на розрахунку відстані x до кожного зразка в базі даних і знаходження мінімальної відстані. Переваги такого методу наступні: ієрархічні структури даних дозволяють значно зменшити кількість обчислень; в базу даних можна додати нові зразки.

– Статистичні методи використовують статистичні дані та інформацію для вирішення задачі розпізнавання. Даний метод визначає чи відноситься об’єкт до певного класу на основі ймовірності чи ні. В загальному це зводиться до визначення ймовірності чи належить об’єкт до певного класу, якщо ознаки цього об’єкту отримали відповідні значення. Прикладом статистичного методу є байєсівські вирішальні правила. Розпізнавання образів застосовуються у тих випадках, коли обрані ознаки характеризуються числовими значеннями, причому ці значення для об’єктів одного класу можуть бути різними. Завдання

розпізнавання образів зводиться, таким чином, до визначення меж областей, що відповідають різним класам. Це один з напрямків теорії розпізнавання образів, в основі якого лежить уявлення про клас розпізнаваних об’єктів як про ряд реалізацій деякої випадкової величини. Завдання визначення значень параметрів, які заважають, є завданням адаптації розпізнавання. При навчанні задається навчальна вибірка, що складається з об’єктів, класи яких вказані. На цій основі, в залежності від того, наскільки детально відомі статистичні характеристики даної моделі, будуються ті чи інші статистичні оцінки самих параметрів або певних функцій. Отримані оцінки потім використовуються в процесі вирішення завдання розпізнавання шляхом їх підстановки замість невідомих значень параметрів. Розглянуті до поточного моменту методи розпізнавання були засновані на навчанні з учителем. Однак навчання без учителя також можливо. При такому підході машина сама повинна визначити структуру класів і приналежність кожного образу певного класу [27].

– Нейронна мережа – це метод параметричної апроксимації, що довів свою корисність в побудові моделей щільності. Нейронні мережі зазвичай апроксимують деяку векторну функцію деякого вхідного вектору поруч рівнів. Кожен рівень формує вектор виходів, кожен з яких представляє собою результат дії певної нелінійної функції. При розпізнаванні методом найближчого сусіда результат методу був повністю визначений еталонними зразками, завантаженими в пам’ять перед обробкою. Тоді як при застосуванні нейронного методу розпізнавання кінцевий результат заздалегідь невідомий. Алгоритми нейронних мереж самі вирішують задачу вибору ознак. З цим пов’язаний один з недоліків – перенавчання.

Розглянемо більш детальніше найпоширеніші методи.

Метод опорних векторів виступає одним із ключових методів, що широко застосовується у сфері розпізнавання об’єктів, має на увазі використання класичних моделей-класифікаторів. Цей підхід представлений великою кількістю моделей, серед яких найбільш широко використовується регресивна модель, штучна нейронна мережа (багатошаровий перцептрон), метод опорних

векторів, а також метод прийняття рішень і моделі ансамблі, що являють собою поєднання деяких перерахованих моделей [33].

Модель багатошарового перцептрона є сукупністю штучних нейронів, об'єднаних в рівні, які задані в ієрархічному порядку. Штучний нейрон виступає як специфічна модель біологічного нейрона, який пересічна людина сприймає за прототип нервової клітини, що має один або більше двох входів, в той же час одним виходом і можливістю активації. Разом з тим, кожен вхід штучного нейрона має асоційований коефіцієнт або вагу.

Як відзначають дослідники, такі багатошарові нейронні мережі здатні інкапсулювати будь-яку математичну функцію за допомогою довільного набору нейронів. Досить часто постає проблемним формулювання аналітичної норми класифікації зображень за категоріями, можливість навчатися на базі вибірки робить спорідненими з нейронними мережами моделі, призначені для розпізнавання природних зображень навколишнього світу, що відрізняються нечіткою структурою і множною варіацій в межах класу.

Метод опорних векторів має деякі переваги і недоліки по відношенню до використання багатошарових перцептронів:

– Багатошаровий перцептрон є моделлю з множиною прихованих параметрів, що залежать від числа нейронів мережі. Параметризована модель в перспективі наділена здатністю до інкапсуляції більш складних функцій, але разом з тим ставить умову більшої кількості часу і обчислювальних ресурсів для навчання та налаштування параметрів. Метод опорних векторів застосовує вектори, відібрані з навчальної вибірки, при цьому кількість параметрів обмежено розміром вибірки, а на практиці може бути проріджені за використання інженерії ознак [26].

– Нейронна мережа вимагає мінімальних обчислювальних ресурсів для роботи в режимі розпізнавання (передбачення категорій). Метод опорних векторів в деяких випадках будує передбачення помітно повільніше, у випадку наявності більшого числа векторів, у порівняні із кількістю вибірки.

– Нейронна мережа наочно показує розширені функціональні можливості до дистанційного навчання, порівнюючи його із методом опорних

векторів, у випадку коли розмірне значення вибірки не закріплене ніяким чином і поповнюється за рахунок отримання нових даних.

У більшості існуючих додатків алгоритмів розпізнавання і машинного навчання на сьогоднішній день переважає метод опорних векторів за рахунок скорочення часу навчання та стійкості до локального мінімуму. Метод опорних векторів також широко використовується для розпізнавання зображень, таких як людські особи, демонструючи високу точність розпізнавання (80-85% успішно ідентифікованих зображень). Характерна риса завдання розпізнавання зображень полягає в тому, що інформаційні дані, які представляють собою візуальні сигнали, демонструють достатньо низьку інформаційну ємність – тобто, більша частина точок вихідного зображення не містить інформації, що впливає на розпізнавання. Для класичних методів розпізнавання об’єктів характерним є прямопропорційна залежність між розмірністю даних вибірки та часовим відрізком навчання, а також рівнем збіжності в процесі оптимізації моделі.

Класичний метод головних компонент, але разом з тим, непридатний для більшості зображень через обчислювальної складності побудови коваріційної матриці. Турк і Пентланд в 1991 р запропонували алгоритм розпізнавання *Eigenfaces*, де використовували альтернативний, прийнятний для сучасних комп’ютерів метод розрахунку власних векторів. В їхньому прикладі метод використовувався на фронтальних фотографіях обличь людини (рис. 2.1). Підтверджуючи положення про те, що розмірність зображення може бути значно знижена, зберігаючи при цьому досить інформації для успішного розпізнавання людиною, вони показали, що кожну з одиниць вибірки можна уявити за допомогою обмеженого набору головних компонент [18].

Рисунок 2.1 – Приклад головних компонентів методу *Eigenfaces*

Для розпізнавання тестові зображення проектувалися на базі з вибраних головних компонент, тобто представлялися у вигляді лінійної суми складових. Використання *Eigenfaces* дозволяло ефективно розпізнавати обличчя при різному освітленні і надавало деякої стійкості до орієнтації; проте, алгоритм погано працював на обличчях різного розміру (варіації масштабу). Окрім перерахованого, метод головних компонент мав й інші обмеження, які сприяли появі нових методів представлення зображень.

У своїй праці Б. Ольшозен наочно показав, що алгоритм, названий розрідженим кодуванням, наділений можливістю ефективніше демонструвати внутрішню структуру зображення і об’єктів в ньому, при цьому показуючи конкретні специфічні властивості, досить схожі з характерними ознаками клітин головного мозку людини. Розрідженість результуючого одержаного уявлення передбачається тим, що більшість компонентів для окремо взятого зображення, будуть рівнятись нулю [21] .

Існують різноманітні закономірні алгоритми пошуку розрідженого коду для вибірки зображень, таких як ортогональне узгоджене переслідування чи застосування специфічних нейронних мереж тощо. Перевага згаданого способу, на противагу методу головних компонент виражається в тому, що компоненти, отримані за допомогою другого методу, практично у будь-якому випадку представляють собою лінійні перетворення вхідних даних, тоді як у разі розрідженого коду елементи мають місце бути нелінійними, інкапсулюючи таким чином, більш складні представлення даних. Також широко використовується клас алгоритмів, здатний формувати цілісне представлення об’єктів – обмежені машини Больцмана.

Обмежена машина Больцмана являє собою нейронну мережу, яка навчається формуванню деякій ймовірності диференціації власних входів. Вони представляють собою модифікацію класичних машин Больцмана, які виступають доповненим варіантом мереж Хопфілда. Зв’язки між нейронами згаданої мережі являють собою двочастковий граф, де одна частина відповідає вхідному рівню мережі, а друга – прихованому. Кожен вхідний нейрон з’єднаний з усіма прихованими нейронами за допомогою симетричних зв’язків, при цьому

нейрони в межах кожної частини не мають зв’язків один з одним (на відміну від класичних, “необмежених” де такі зв’язки можливі).

Істотною перевагою методів, які формуються з допомогою розрідженого кодування і обмеженої машини Больцмана, порівнюючи його із методом головних компонент є їх нелінійність, що дозволяє досліджувати способи нарощування таких репрезентативних моделей. Цей метод більш поширений під назвою глибокого навчання і окреслений різкою максимізацією точності розпізнавання в достатньо значних галузях машинного навчання, включаючи розпізнавання зображень. Фундаментальним виступає припущення про те, що теорії, відповідно яким навчаються репрезентативні моделі, мають ієрархічну природу [38].

Для розпізнавання зображень успішно застосовувалися глибокі моделі, що складаються з обмежених машин Больцмана – так звані глибокі мережі довіри. Практичне застосування ієрархічних тверджень надає можливість такій моделі навчатися досить складним, масштабним об’єктам, передбачаючи додаткові рівні стійкості до інваріантних перетворень. Так, глибока модель, навчена на базі людських обличь, здатна розпізнавати значно більш суттєві викривлення, ніж модель *Eigenfaces*, що включають в себе обертання об’єкта в межах обмежених кутів. В цілому глибокі моделі забезпечують більш гнучкі і багаті представлення, які підходять для об’єктів зі складною структурою. Проблемні питання, що виникли в процесі використання моделей, що формують цілісні репрезентації, в значній мірі сприяли активному розвитку нового класу алгоритмів, що застосовують локальні ознаки зображень.

Згорткові мережі – один з найбільш успішних існуючих на сьогоднішній день алгоритмів розпізнавання зображень. Недоліками можна назвати труднощі при обробці об’єктів малого розміру і обмежена можливість вирішувати спотворення. При цьому згорткові мережі досить легко вирішують питання високоточного розпізнавання, які викликають труднощі у пересічних людей – наприклад, ідентифікація окремих моделей машин чи порід собак, й інші завдання, що вимагають виділення вузько специфічних ознак.

Згорткова нейронна мережа може бути описана наступною формулою,

$$(f * g)[m,n] = \Sigma_f[m-k, n-L] * g[k,l],$$

де f – вихідна матриця зображення; g – ядро (матриця) згортки.

Неформально цю операцію можна описати наступним чином – вікном розміру ядра g проходимо із заданим кроком (зазвичай 1) все зображення f на кожному кроці поелементно множимо на вміст вікна на ядро g, результат підсумовується і записується в таблицю результату.

Однією з основних особливостей згорткових мереж є той факт, що така модель не має інформаційних даних про те, яким саме чином локалізований зображений розшукуваний об'єкт – його конкретне місцезнаходження та орієнтація у просторі. При цьому в розв'язанні прикладних задач управління й обробки інформації, знання параметрів локалізації є необхідною умовою – в залежності від розташування або пози об'єкта система обробки інформації може класифікувати зображення по-різному відповідно до покладених на неї завдань [22].

Архітектура згорткових мереж не передбачає стійкості до інших перетворень, таких як дзеркальне відображення і масштабування. Для вирішення цієї проблеми, як правило, використовуються евристичні методи (вирівнювання горизонту зображення, використання просторових пірамід тощо). Структуру згорткової мережі можна образити наступним чином (рис. 2.2).

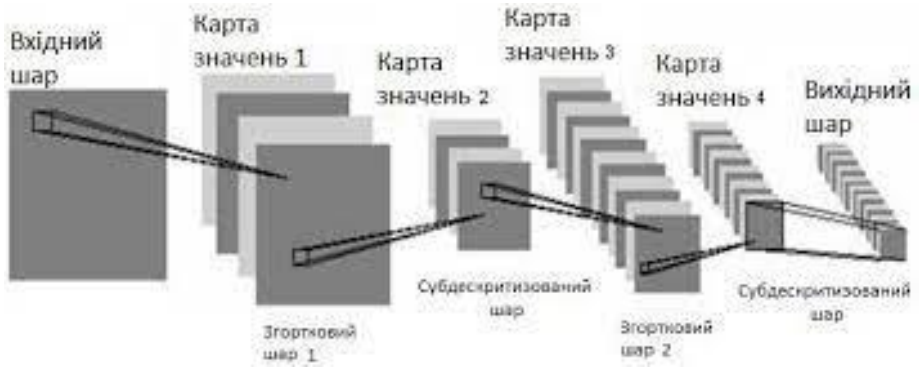

Рисунок 2.2 – Структура згорткової мережі

Альтернативні підходи до виділення локальних ознак включають в себе методи класичного комп'ютерного зору, які не використовують навчальні моделі. Ці методи здійснюють пошук на зображенні характерних ділянок, що

відповідають алгоритмічно явно заданим умовам (рис. 2.3). Серед них виділяються:

– Виявлення країв, що визначається як ділянка зображення, яка представляє собою «кордон» між двома контрастними регіонами, помітна людському оку. Виділення меж часто використовується як попередній етап обробки зображень в інших алгоритмах розпізнавання, в тому числі – згорткових мережах.

– Виявлення кутів або “точок інтересу”. До цієї групи відносяться алгоритми, що виділяють локальні ділянки зображення, максимально чутливі до змін. Традиційно ця група алгоритмів використовувалася для відшукання кутів між прямими лініями, але в даний момент розглядаються також будь-які точки з високим значенням кривизни.

– Виявлення ділянок неоднорідності. Під ділянками неоднорідності, на відміну від кутів, розуміються деякі безперервні регіони зображення, які відрізняються за значеннями кольору або інтенсивності від навколишнього фону, і при цьому схожі між собою.

Перераховані локальні ознаки широко використовуються в задачах візуального трекінгу і стеження за об’єктом, але в чистому вигляді непридатні для задачі розпізнавання в силу своєї недискримінативної природи [17].

Рисунок 2.3 – Метод визначення локальних ознак

Методи виділення локальних ознак дозволяють справлятися із деякими класами проблем розпізнавання зображень, забезпечуючи стійкість до оклюзії, знижуючи обчислювальне навантаження при обробці зображень високої

розмірності і дозволяючи формувати інваріантні ознаки для виявлення об’єктів під дією інваріантних перетворень.

Констеляційна модель – це ймовірна породжуюча модель для класифікації і розпізнавання об’єктів в області комп’ютерного зору. Така модель формує представлення об’єкта від більшості компонентів зображення, обраних з урахуванням виокремлених геометричних закономірностей, при цьому враховується розташування компонентів на відміну від інших моделей, що використовують локальні ознаки, які в загальному випадку навмисно ігнорують інформацію щодо розташування ознак зображення в просторі.

Дослідження повноти і точності розпізнавання за допомогою констеляційної моделі показують, що така модель здатна ефективно справлятися з великими вибірками, що включають в себе категорії людських обличь, мотоциклів, літаків, плямистих кішок тощо. Для кожного з таких класів згадана модель складає презентацію об’єкта з використанням форми і просторового розташування його складових частин, відповідних “істотним” атрибутам класу. Подібна гнучкість допомагає моделі успішно обробляти як “жорсткі”, фіксовані об’єкти, так і аморфні, здатні до змін [24].

Така модель вимагає значних обчислюваних ресурсів зі збільшенням кількості частин, які складають структуру моделі. Коректна констеляційна модель здатна ідентифікувати окремі частини об’єкту та прогнозувати можливість того, що зображення належить до необхідного класу (рис. 2.4).

Рисунок 2.4 – Приклад успішно ідентифікованого об’єкту

Д. Фергюсом і Л. Фей-Фей був запропонований модифікований варіант моделі під назвою “модель-зірка”, для якої асимптотика процесу навчання знижена до 0. Це значення дозволяє використовувати значно більше число використовуваних компонентів моделі й потенційних активацій ознак в одному зображенні. Констеляційна модель має істотну відмінність від розглянутих раніше моделей. На відміну від цілісних і локально-інваріантних уявлень, вона базується на деякому наборі апріорних припущень про те, як організована структура об’єкта, а саме, висловлюється припущення про те, що будь-який досить складний для розгляду в задачі розпізнавання об’єкт, повинен складатися з просторово-зв’язкових локальних складових частин [31].

Актуальним питанням при розпізнаванні зображень є оцінка ефективності роботи методу розпізнавання. Для отримання чисельного значення оцінки широко використовуються як загальні методи математичної статистики, так і специфічні показники, що застосовуються для оцінки алгоритмів машинного навчання. Однією із найпростіших метрик оцінки ефективності виступає відсоткова частка коректно розпізнаних зображень. Коректним розпізнанням

вважається отримання на виході алгоритму класу, відповідного попередньо заданому класу. Для оцінки використовується вибірка, спроектована аналогічно навчальній вибірці, але така, що містить ображення, до яких алгоритм не міг мати доступ в процесі навчання. Розглянутий показник є найбільш узагальненим і підходить для більшості завдань розпізнавання з обмеженою кількістю класів .

У завданнях, де кількість класів не фіксоване, і завдання розпізнавання являє собою завдання ідентифікації об'єкта певної категорії серед множини інших, потенційно необмежених категорій, замість неї застосовуються такі показники, як точність і повнота оцінки. Їх використання дозволяє розрізнити хибно-позитивні (класифікатор прийняв позитивне рішення за зображенням, що не містить шуканого об'єкта) і помилково-негативні (класифікатор не пізнав об'єкт на зображенні, де він був присутній) помилки розпізнавання, або помилки першого і другого роду. Таким чином, точність оцінки в межах класу представляє собою елемент зображення, який дійсно належать даному класу щодо всіх зображень, які система віднесла до цього класу[36].

Показники точності і повноти широко використовуються в області обробки інформації, і як правило, розраховуються спільно. При цьому існує кілька методик зіставлення двох показників:

– кожен показник враховується індивідуально;

– для випадків, коли між показниками може спостерігатися залежність, проводиться оцінка одного з показників при фіксації;

– обидва показники можуть бути скомбіновані в один.

Крім перерахованих показників, важливу роль в оцінці ефективності методів розпізнавання грають показники, що дозволяють оцінити метрики роботи алгоритмів, не пов'язаних безпосередньо з прийняттям класифікуючих рішень. Серед цих показників можна виділити розмір інтервалу для вибірки зображень, оцінка числа інтеграцій і часу роботи алгоритму при навчанні, необхідних для досягнення збіжності.

2.3 Проектування системи розпізнавання зображень

Реальна система розпізнавання – це функціонально взаємопов’язана сукупність методів і технічних засобів, що здійснює процес синтезу і аналізу образів.

Перш ніж приступати до побудови системи розпізнавання необхідно проаналізувати всю доступну інформацію про об’єкти дослідження і вирішити наступні питання.

1. Якими загальними характеристиками і властивостями володіють об’єкти дослідження і чим вони відрізняються.

2. Якщо необхідні характеристики можуть бути отримані в результаті вимірювань, яка точність цих вимірів.

3. Чи існує відповідна модель (моделі) для формального опису та аналізу даних характеристик.

На підставі проведених досліджень визначається тип і структура системи розпізнавання. За фізичною природою характеристик-ознак образів системи розпізнавання поділяються на прості та складні. Прикладом простої системи розпізнавання в завданнях обробки даних є класифікація пікселів багатозонального сканера або оцифрованих спектрозональних зображень. В цьому випадку ознаками є спектральні відбивні властивості об’єктів земної поверхні. Як тільки в процес класифікації залучаються інші типи даних, система стає складною.

Складні системи розпізнавання можуть бути однорівневими і багаторівневими. У однорівневих системах розпізнавання здійснюється на основі одного словника ознак одним алгоритмом розпізнавання. У багаторівневих системах результати розпізнавання, отримані на одному етапі, використовуються в якості вихідних даних на наступному.

Велика частина технологій тематичної обробки даних при розв’язанні прикладних задач реалізується складними багаторівневими системами розпізнавання. Розбиття схеми виконання завдання на рівні називається декомпозицією. Спосіб декомпозиції системи розпізнавання зазвичай пов’язаний з моментами включення в процес розпізнавання фахівця-аналітика даних, тобто з інтерактивним режимом обробки.

Необхідність в інтерактивному режимі найчастіше виникає на етапі синтезу образів через неповноту вихідної інформації або труднощів її формального опису. У багаторівневих системах важливою функцією аналітика даних є оцінка якості розпізнавання на поточному етапі і виникає через це втрати інформації на наступному.

За повнотою вихідної інформації системи розпізнавання поділяються на системи з навчанням, без навчання і самонавчання системи. У системах без навчання передбачається, що наявна інформація і обраний принцип розпізнавання дозволяють безпомилково розділити всі необхідні класи. Типовим прикладом систем без навчання є системи, засновані на принципі порівняння з еталоном (принципі перерахування). Системи без навчання можуть бути побудовані і на принципі кластеризації, коли вектори-образи кожного класу утворюють компактні групи. Якщо наявних даних про об'єкти дослідження недостатньо для їх точного поділу, розпізнавання без навчання призводить до великих помилок. І навпаки, кількість інформації про об'єкти може бути настільки великим, що для виділення потрібних класів немає необхідності використовувати її в повному обсязі. У цих випадках застосовують системи розпізнавання з навчанням.

Системи з навчанням широко використовуються при тематичній обробці даних. Процедура навчання зазвичай виконується фахівцем-аналітиком даних в інтерактивному режимі. Слід відзначити, що будь-які процедури класифікації з навчанням більш трудомісткі і якість розпізнавання іноді більше залежить від особистого досвіду аналітика даних і розуміння ним суті поставленого завдання, ніж від ефективності самого алгоритму розпізнавання. Найчастіше системи з навчанням ґрунтуються так чи інакше на принципі спільності властивостей. При цьому образи не обов'язково повинні бути представлені якоюсь структурою. Це може бути просто множина параметрів-вимірювань, що підкоряються в кожному класі певного закону статистичного розподілу, або безліч якихось атомарних (неподільних) елементів, присутніх в кожному класі в певній комбінації. Такі множини в узагальненій теорії образів називають вільною конфігурацією [37].

Для оцінки якості розпізнавання використовується деякий функціонал, пов’язаний з помилками розпізнавання, який в процесі навчання максимізується або мінімізується. Велике значення при декомпозиції складних систем мають вимоги до точності розпізнавання, тобто до допустимої ймовірності помилок. Якщо в одному алгоритмі розпізнавання використовуються ознаки різної фізичної природи або отримані з принципово різних моделей, то величина помилки може виявитися непередбачуваною. Після декомпозиції складної системи кожен рівень можна розглядати як окрему підсистему розпізнавання зі своїм списком класів, словником ознак і правилами класифікації.

Оцінка ефективності системи розпізнавання виконується для всієї системи в цілому, оскільки вона може істотно залежати від способу декомпозиції задачі. Її можна оцінювати за такими показниками:

- Підсумкова точність розпізнавання.
- Загальний час розпізнавання.
- Вартість розробки і експлуатації системи.

При побудові інтерактивних систем розпізнавання, зокрема, обробки даних, в оцінку ефективності системи можуть бути включені витрати на підготовку фахівців-аналітиків, які беруть участь в процесі розпізнавання, оскільки від їх кваліфікації часто залежить підсумкова точність і загальний час виконання завдання.

Процедура побудови штучних нейронних мереж, які використовуються в задачах класифікації, включає наступні етапи[9].

Етап 1. Визначення типу образів, що розпізнаються (фотографія, відео, сукупність числових характеристик і т.п.), набору вимірюваних параметрів образів, інформативних з точки зору розпізнавання, і класів розпізнаваних образів.

Етап 2. Вибір способу перетворення (застосування фільтрів, придушення шумів, сегментація тощо). Реальні показники можуть відрізнятися від образів у вхідному векторі числових величин та подання набору класів образів у вигляді вектора вихідних величин. Ефективність нейромережевої моделі підвищується,

якщо діапазони зміни величин вхідного і вихідного векторів будуть нормалізовані.

Етап 3. Проектування архітектури штучної нейронної мережі:

– визначення кількості шарів. У мережі повинно бути, як мінімум, два прошарки: вхідний і вихідний. Кількість прихованих шарів визначається, як правило, експертним або досвідченим шляхом;

– визначення кількості нейронів в кожному шарі. Кількість нейронів у вхідному шарі відповідає кількості перетворених вимірюваних параметрів розпізнаються образів. Кількість нейронів у вихідному шарі, як правило, відповідає кількості класів розпізнаваних образів. Визначення кількості нейронів в кожному прихованому шарі є неформальною проблемою, при вирішенні якої можна використовувати евристичне правило: число нейронів в наступному шарі повинно бути в два рази менше, ніж в попередньому;

– вибір типу зв'язків між нейронами. Залежить від специфіки розв'язуваної задачі;

– вибір функції активації. Залежить від специфіки розв'язуваної задачі.

Етап 4. Навчання мережі – уточнення значень вагових коефіцієнтів зв'язків на основі багаторазового прогону навчальних прикладів через мережу. Кожному навчальному образу відповідають певні вектори вхідних і вихідних величин. Навчальні приклади, як правило, представляють собою еталонні (ідеальні) представлення образів, а також їх незначні модифікації. Вони повинні бути підібрані для кожного класу образів. Існує емпіричне правило, яке встановлює рекомендоване співвідношення між кількістю навчальних прикладів і числом зв'язків в нейронній мережі.

Етап 5. Тестування мережі за допомогою контрольного набору прикладів (образів) для оцінки якості обраної архітектури та ступеня навчання. Контрольний набір повинен містити, як мінімум, по одному прикладу для кожного класу образів і включати в себе як еталонні, так і значно спотворені образи.

Етапи з третього по п'ятий можуть спільно застосовуватися для уточнення архітектури мережі на основі конструктивного або деструктивного підходу.

Відповідно до першого підходу навчання починається на мережі невеликого розміру, який поступово збільшується до досягнення необхідної точності за результатами тестування. Деструктивний підхід базується на принципі “проріджування дерева”, відповідно до якого з мережі з явно надмірною об’ємом поступово видаляють “зайві” шари, нейрони і примикають до них зв’язку. Цей підхід дає можливість досліджувати вплив елементів, що видаляються, на точність розпізнавання мережі [33].

Для реалізації зручної та ефективної інформаційної системи необхідно визначитися із завданнями, які буде вирішувати дана система. Головні завдання будуть наступні:

– виконання розпізнавання об’єктів з зображення;

– фіксація ідентифікованих об’єктів до бази даних та додавання, редагування та видалення даних у таблиці бази даних;

– ведення ефективного пошуку необхідних даних користувачеві в залежності від запитів;

– створення можливості зручного перегляду існуючих даних в базі даних;

– реалізація можливості експорту даних в файл Excel;

– комфортний та зручний інтерфейс користувача для завантаження зображень декількома варіантами та збереження результатів.

При розробці інформаційної системи доцільно дотримуватися такого порядку:

– вивчення і перевірка специфікації системи,

– вибір мови програмування;

– вибір алгоритму і структури даних;

– програмування (кодування) системи;

– шліфування тексту системи;

– перевірка системи;

– компіляція системи.

Перший крок розробки програмного модуля в деякій мірі є суміжний до процедури контролю структури програми знизу: вивчаючи специфікацію

модуля, необхідно переконатися, що специфікація зрозуміла і достатня для розробки цього модуля. На завершення цього кроку вибирається мова програмування: хоча мова програмування може бути вже визначена для всієї системи, все ж в деяких випадках (якщо система програмування це допускає) може бути вибрана інша мова, яка більше підходить для реалізації даного модуля. На другому кроці розробки програмного модуля необхідно з'ясувати, чи відомі та чи є вже якісь алгоритми для вирішення поставленого завдання. І якщо знайдеться відповідний алгоритм, то доцільно їм скористатися. Вибір відповідних структур даних, які будуть використовуватися при виконанні модулем своїх функцій, в значній мірі визначає логіку і якісні показники розроблюваної системи, тому його слід розглядати як вельми відповідальне рішення. На третьому кроці здійснюється побудова тексту модуля обраною мовою програмування. Велика кількість всіляких деталей, які повинні бути враховані при реалізації функцій, зазначених у специфікації модуля, легко можуть привести до створення досить заплутаного тексту, що містить масу помилок і неточностей. Шукати помилки в такому модулі і вносити в нього необхідні зміни може виявитися досить трудомістким завданням. Наступний крок розробки інформаційної системи пов'язаний з приведенням тексту модуля в завершеному вигляді відповідно до специфікації якості програмного середовища. При програмуванні модуля необхідно приділяти особливу увагу правильності реалізації функцій модуля, залишаючи недопрацьованими коментарі і допускаючи деякі порушення вимог до стилю програми. При шліфуванні тексту модуля необхідно відредагувати наявні в тексті коментарі і, можливо, включити в нього додаткові коментарі з метою забезпечити необхідні примітиви якості. З цією ж метою проводиться редагування тексту програми для виконання стилістичних вимог. Крок перевірки модуля являє собою ручну перевірку внутрішньої логіки модуля до початку його налагодження (використовує виконання його на комп'ютері), реалізує загальний принцип, сформульований для обговорюваної технології програмування. Останній крок розробки модуля означає завершення перевірки модуля (за допомогою компілятора) і перехід до процесу налагодження модуля[10].

2.4. Проектування структури системи, вибір мови програмування та середовища розробки

Виходячи з поставлених задач, формується функціональна схема з зображенням взаємодії блоків інформаційної системи. Функціональна схема – це схема взаємодії компонентів програмного забезпечення з описом інформаційних потоків, складу даних у потоках і вказівкою використовуваних файлів і пристроїв (рис. 2.5).

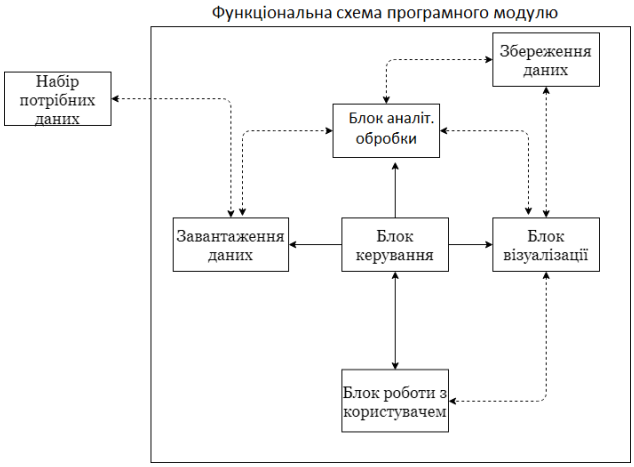

Рисунок 2.5 – Функціональна схема програмного модулю

Для виконання завдання була розроблена таблиця бази даних, наведена на рис. 2.6.

Рисунок 2.6 – Таблиця бази даних

Таблиця містить чотири ключових поля, що забезпечують роботу інформаційної системи.

Поле *id_object* – ідентифікується як унікальний порядковий номер розпізнаного об’єкта на зображенні.

Поле *nameObject* – назва знайденого об’єкта. Назва формується із бібліотеки Yolo.

Поле *nameFile* – файл, який знаходиться на елементі pictureBox.

Поле timeRecjgnition – окреслює точний час процесу розпізнавання (враховує поточну дату та точний час).

Для реалізації програмної частини обрано мову програмування третього покоління *C#. C#* це об’єктно-орієнтована мова програмування з безпечною системою типізації для платформи *.NET*. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою *Microsoft*. Синтаксис *C#* схожий із *Java* та *C++* . Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі *XML*. Рейтинг використання мов програмування зображено на рис. 2.7 [7].

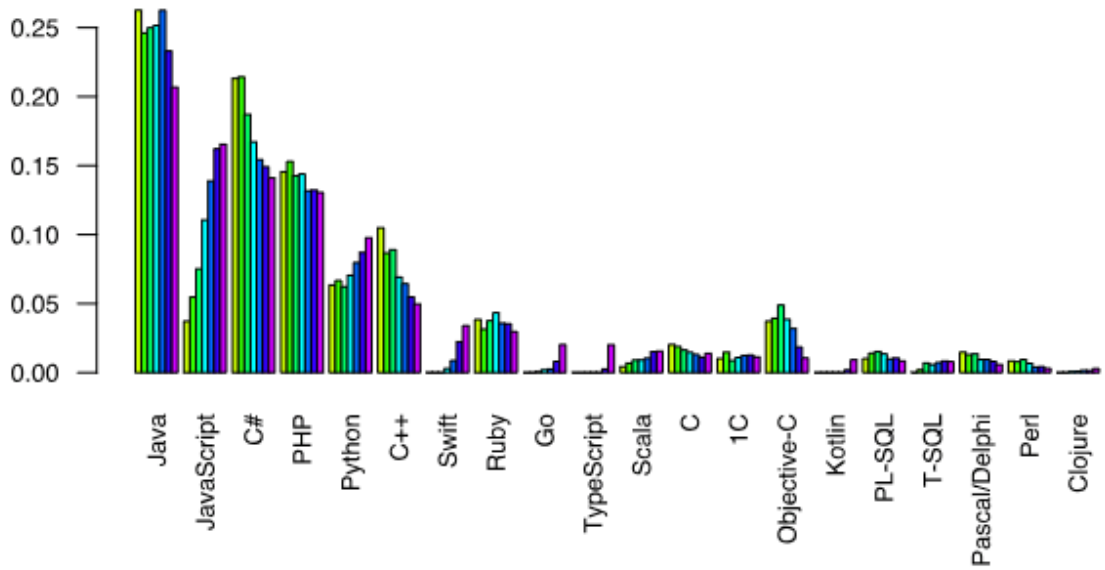

Рисунок 2.7 – Рейтинг мов програмування

Мова програмування *C#* (рис. 2.8) здебільшого застосовується для формування корпоративного програмного забезпечення, фінансових проектів, наприклад для бірж та банків , зокрема хмарних сервісів мобільних додатків.

Рисунок 2.8 – Логотип мови програмування *C#*

На відміну від *Java C#* легше піддається взаємодії із кодом програм, які написані на інших програмних мовах. Варто також зазначити, що на *C#* досить часто пишуться розширення для інших мов програмування, що застосовуються в якості прошарку між бібліотекою *C#* і мовою, можливості якої планується розширити (під конкретні цілі).

C# досить часто застосовується в розробці ігор на *Unity* – одному із найпоширеніших ігрових двигунів. Отже, можна зробити висновок, що за допомогою *C#* створювалися сотні ігор, включаючи найпопулярніші. *C#* також чудово підходить під роботу системою *embedded*. *Embedded system* – спеціалізована комп'ютерна система, призначена для виконання обмеженої кількості функцій.

На відміну від інших мов програмування, *C#* є високорівневою, що означає, що вона в деякій мірі має схожі риси із англійською. Мова програмування *C#* має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі *XML*.

В якості середовища розробки обрано *Microsoft Visual Studio* (рис. 2.9) – серія продуктів компанії *Microsoft*, які включають інтегроване середовище розробки програмного забезпечення та перелік інших інструментальних засобів. Ці продукти дають можливість розробляти консольні програми, програми з графічним інтерфейсом, в тому числі з підтримкою технології *Windows Forms*, а також веб-сайти, веб-застосунки тощо [18].

Рисунок 2.9 – Програмне середовище *Visual Studio*

У якості програмної технології обрано платформу *.Net Framework*. *Microsoft .Net Framework* є так званої програмною платформою. У загальних рисах можна провести аналогію з відеофайлами, які не будуть відтворюватися якщо в системі не встановлений потрібний кодек. У даному випадку відеофайл – це програма, написана з використанням технології *.Net*, а кодек – це сама платформа *Microsoft .Net Framework*. Зроблено це для того, щоб розробник міг максимально абстрагуватися від системного оточення на комп'ютері користувача. Його не повинно хвилювати, яка операційна система встановлена, яка розрядність у процесора - 32-х або 64-бітна, яка у нього архітектура тощо. Для запуску програми досить, щоб під цю систему існувала і була встановлена реалізація *.Net Framework*. За допомогою платформи *.NET* для створення графічних інтерфейсів на практиці використовуються різні технології – *Window Forms*, *WPF*, додатки для магазину *Windows Store*. Однак найпростішою та найбільш зручною платформою досі залишається *Window Forms*.

Windows Forms використовується в *Microsoft.NET* для створення додатків, які доповнені графічним інтерфейсом. Ґрунтується він на *.NET Framework class library* і має більш зручну та досконалу в роботі модель програмування, ніж, для прикладу, програмні інтерфейси *Win32 API* або *MFC*.

Windows Forms – це набір різних, керованих бібліотек, за допомогою яких можна виконати всі необхідні для віконного додатка дії, починаючи від обміну повідомленнями з операційною системою для відстеження будь-яких подій клієнтського вікна, і закінчуючи діалоговими системами, зв'язком з іншими комп'ютерами у мережі й багатьма іншими можливостями. У такому випадку, під формою розуміється видима поверхня вікна, що включає інформацію для

кінцевого користувача, і містить в собі набір елементів керування для роботи із представленими даними або взаємодії з користувачем [22].

2.5. Архітектурні особливості інформаційної системи

Архітектурний шаблон визначає структурну організацію системи. Він надає набір визначених підсистем, їх обов'язки, а також принципи для організації взаємодії між ними. А шаблон проектування описує схему взаємодії компонентів в рамках підсистеми, їх відносини та ін.

Алгоритм розпізнавання зображень є ключовим елементом методу, представленого в даному дослідженні, здійснює процедуру класифікації зображень за категоріям з використанням попередньо навченої моделі. Алгоритм розпізнавання складається з послідовної активації детекторів моделі.

Побудова системи розпізнавання вимагає попереднього аналізу всієї доступної інформації про досліджувані об’єкти. На цьому попередньому етапі необхідно:

– зібрати всі характеристики досліджуваних об’єктів, які можна витягти з доступних даних, що мають відношення до розв’язуваної задачі;

– проаналізувати можливості формалізованого опису семантичних і непрямих ознак об’єктів;

– сформувати повний набір формалізованих ознак об’єктів дослідження;

– визначити, за якими параметрами розрізняються об’єкти підлягають виділенню тематичних категорій, і за якими характеристиками подібні об’єкти кожної категорії;

– вивчити поведінку отриманих ознак, які підлягають виділенню на кожній із тематичної категорії (класі) об’єктів і визначити набір найбільш інформативних ознак, тобто дозволяють надійно розділяти необхідні класи;

– проаналізувати можливість застосування до обраних ознак, описаних вище, принципів розпізнавання та розробити загальну схему вирішення задачі [26].

Якщо наявний набір ознак і перелік класів не дозволяє вирішити задачу одним алгоритмом, необхідно розбити задачу на окремі процедури і для кожної визначити принцип розпізнавання, тобто виконати декомпозицію системи розпізнавання. Як зазначено вище, система розпізнавання на кожному етапі включає дві основні функції: процес синтезу образів і процес аналізу образів. До процесу синтезу образів відноситься:

- формування набору ознак;
- визначення переліку класів;
- опис класів в обраній системі ознак;
- оптимізація набору ознак щодо обраного переліку класів.

Остання процедура потрібна при надмірності вихідних даних, зокрема, мультиспектральних і особливо гіперспектральних зображень. Ці дані призначені для вирішення широкого кола тематичних завдань [29].

Оптимізація набору ознак, яку іноді називають виділенням ознак, дозволяє значно підвищити ефективність системи розпізнавання. При цьому в поняття ефективності системи включається:

- якість результату розпізнавання;
- часові витрати на його отримання;
- витрати на розробку системи;
- витрати на навчання операторів – аналітиків даних.

Якість результату розпізнавання має оцінюватися окремо для кожного алгоритму. Для завдань тематичного дешифрування зображень воно включає два взаємопов’язані поняття: точність і достовірність. Завжди необхідно переконатися, що на аналізованому безлічі образів немає інших тематичних категорій з аналогічним набором ознак. В іншому випадку це неминуче відіб’ється і на точності класифікації. Проте, точність класифікації може залежати і від інших, в основному випадкових, факторів: наприклад, коли суміш двох тематичних класів або наявність стороннього об’єкта в межах пікселя зображення дає ознака, характерний для якихось інших класів. Тому на практиці

підсумкова точність класифікації ніколи не досягає 100%. Допустимою вважається точність класифікації від 70% і вище [34].

Аналіз образів це, власне, сама процедура розпізнавання, тобто сукупність правил прийняття рішення про віднесення образу до класу.

Розпізнавання образів – завдання пошуку та визначення об'єкта та його характерних властивостей зображенню та віднесення його до множини образів за визначеним правилом. Етапи процедури розпізнавання:

– сприйняття образу – відбувається шляхом отримання показників характерних властивостей об’єкта; на цьому етапі відбувається нормалізація;

– індексація – полягає у виділенні характеристик, цей крок полягає у вимірюванні властивостей об’єкта, притаманних лише йому;

– сегментація;

– класифікація – безпосередній процес прийняття рішення.

Для реалізації запропонованого методу створено алгоритм вирішення задачі розпізнавання образів (рис. 2.10).

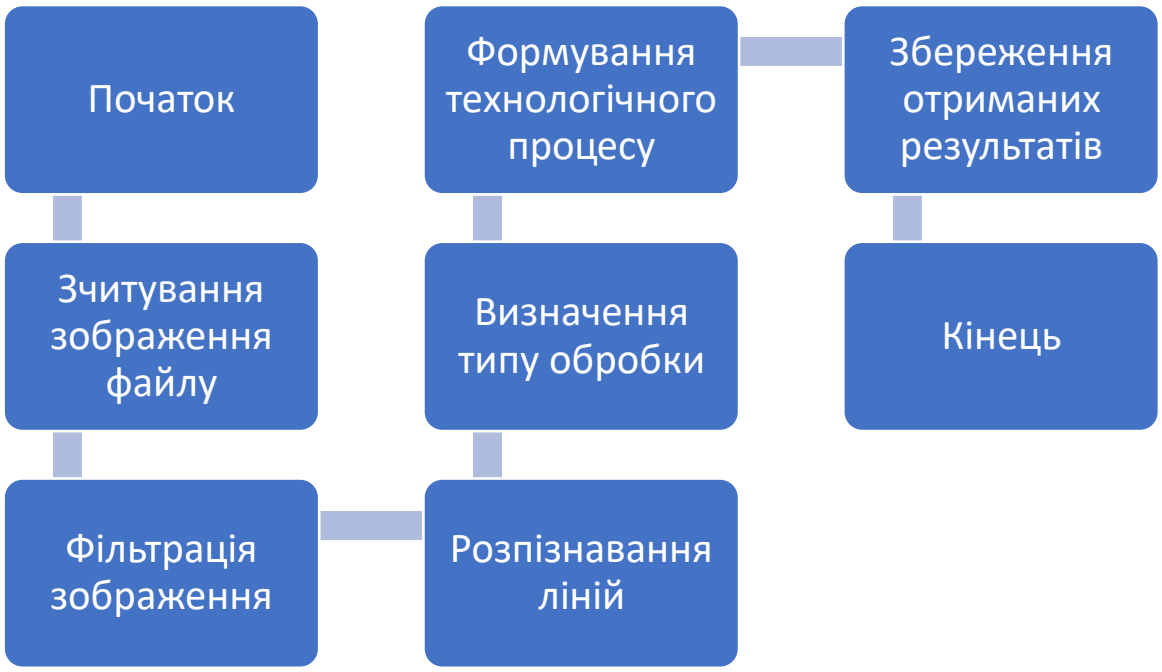

Рисунок 2.10 – Алгоритм методики розпізнавання образів

На схемі показано алгоритм методики розпізнавання образів:

1) крок перший – ввімкнення програми;

2) крок другий – зчитування зображення з файлу у зручному доступному форматі;

3) крок третій – фільтрація зображення – полягає в удосконаленні ображення для полегшення процесу ідентифікації (згаданий процес сегментації тощо);

4) крок четвертий – розпізнавання ліній, за допомогою бібліотеки Yolo;

5) крок п’ятий – оцінка вхідного зображення і визначення типу обробки деталі використовуючи умовні оператори та присвоєння відповідного типу кожному елементу зображення;

6) крок шостий – формування технологічного процесу .

7) крок сьомий – отримання результату технологічного процесу обробки;

8) крок восьмий – кінець роботи програми.

Розглянемо два головних етапи процедури розпізнавання – нормалізацію та сегментацію. Сегментація зазвичай розуміється як процес пошуку однорідних областей на зображенні. Цей етап дуже важкий і в загальному вигляді не алгоритмізований до кінця для довільних зображень. Найбільш поширені методи сегментації, засновані на визначенні однорідних яскравостей (кольорів) або однорідностей типу текстур.

При існуванні стабільних відмінностей в яскравості окремих областей поля зору застосовуються порогові методи. Методи нарощування областей ефективні при наявності стійкої зв’язності всередині окремих сегментів. Метод виділення кордонів добре застосовувати, якщо межі досить чіткі та стабільні. Перераховані методи служать для виділення сегментів за критерієм однорідних характеристик яскравості.

Всі методи дуже прийнятні з точки зору обчислювальних витрат. Однак, для кожного з них характерна неоднозначність розмітки отворів в реальних ситуаціях через необхідність застосування евристик (наприклад, вибір цифрових масок тощо). Заслуговує на увагу в зв’язку з цим запропонований метод розмітки, заснований на комбінації різних прийомів для зниження невизначеності.

Вважаємо доцільним розглянути методи сегментації (рис. 2.11).

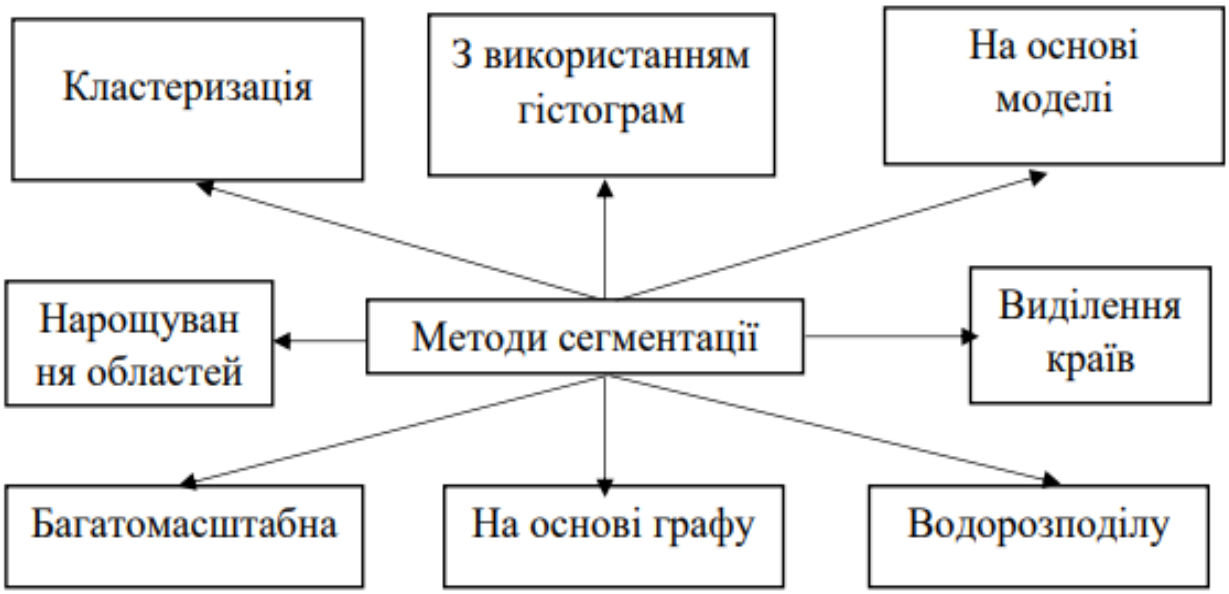

Рисунок 2.11 – Методи сегментації

Для опису та сегментації властивостей зображень, саме, однорідності, шорсткості, регулярності, застосовують текстурні методи, що умовно поділяють на дві категорії: статистичні та структурні [31]. Методи нормалізації при розпізнаванні займають проміжне місце між сигнальними кореляційними і ознаковими алгоритмами. При нормалізації зображення не “втрачається”, а тільки заміщається зображенням того ж класу еквівалентності. У той же час, на відміну від кореляційних методів, безліч вхідних зображень замінюється безліччю нормалізованих зображень. Кожне нормалізоване зображення, в загальному випадку, знаходиться набагато ближче до свого ідеалу (з позиції групових перетворень), що значно скорочує кількість кореляцій на завершальному етапі розпізнавання.

Суть нормалізації полягає в автоматичному обчисленні невідомих параметрів перетворень, яким піддані вхідні зображення, і подальшому приведенні їх до еталонного вигляду. Процедура перетворень проводиться за допомогою операторів нормалізації (нормалізаторів), а обчислення параметрів виконується функціоналами, діючими на множині зображень.

Нормалізація проектних перетворень є складним і ще недостатньою мірою вирішеним завданням. Для найпростіших проективних перетворень типу

перспективи, отримані відповідні нелінійні функціонали типу відносини моментів. Приведення до єдиного масштабу забезпечується нормуванням кожної змінної на діапазон розкиду її значень. У найпростішому варіанті це лінійне перетворення в одиничний інтервал. Лінійне нормування оптимальне, коли значення змінної щільно заповнюють певний інтервал. Набагато надійніше, тому, орієнтуватися при нормуванні на типові, тобто статистичні характеристики даних, такі як середнє і дисперсія.

Всі вище перераховані методи нормування спрямовані на те, щоб максимізувати ентропію кожного входу (виходу) окремо. Але, взагалі кажучи, можна домогтися набагато більшого, максимізуючи їх спільну ентропію. Існують методи, що дозволяють проводити нормування для всієї сукупності входів.

Процес розпізнавання зображень є складною багатоетапною процедурою. Багатоетапність (ієрархічність) обумовлена тим, що різні завдання обробки насправді тісно пов’язані та якість вирішення однієї з них впливає на вибір методу вирішення інших. Так вибір методу розпізнавання залежить від конкретних умов пред’явлення вхідних зображень, в тому числі характеру фону, інших зображень, несприятливої обстановки і пов’язаний з вибором методів попередньої обробки, сегментації, фільтрації.

Однак, для типових ситуацій, на основі проведеного аналізу, можна запропонувати універсальну ієрархічну структуру розпізнавання. На першому етапі застосовуються найменш трудомісткі ознакові алгоритми для вирішення завдання про нееквівалентності вхідних зображень і еталонів. Вхідна множина зображень при цьому істотно скорочується. На другому етапі залишилися зображення, що піддаються нормалізації. На третьому – нормалізовані зображення класифікуються одним із конструктивних способів, наприклад, кореляційним. За такої структури розпізнавання час вирішення завдань скорочується в сотні разів.

РОЗДІЛ 3
РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1. Аналіз вимог до системи розпізнавання зображень.

Інформаційна система повинна містити такі складові:

– реляційну базу даних, розроблену за допомогою СКБД *MySQL*, яка буде вміщувати в собі інформацію щодо організації інформаційної системи із розпізнавання зображень;

– блок завантаження розробленої бази даних;

– блок візуалізації для відображення графічної обробки даних;

– блок роботи з користувачем для внесення необхідних даних для комфортного користування системою;

– блок керування для взаємодії між іншими блоками інформаційної системи;

– блок збереження результатів аналітичної обробки;

– блок аналітичної обробки даних та внесення результатів обробки до програмного модулю.

Вважаємо доцільним розробити також діаграму прецендентів. Цей вид діаграми служить для того, щоб відобразити відносини між акторами і прецедентами, визначити межі системи, описує функціональні вимоги системи (рис. 3.1).

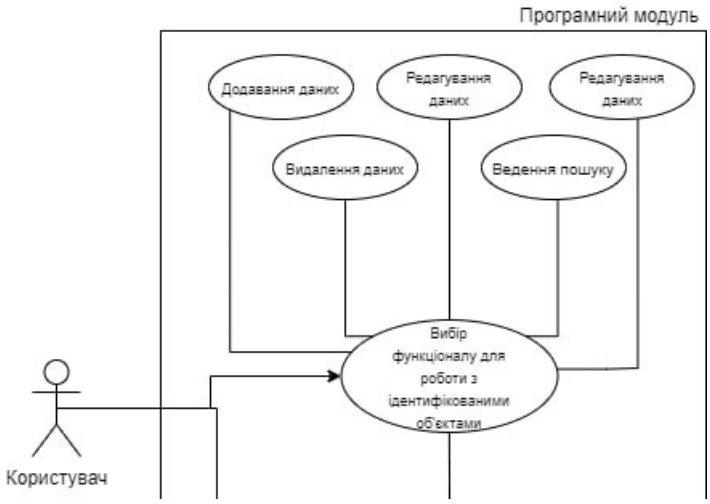

Рисунок 3.1. – Діаграма прецендентів

На діаграмі прецендентів зображено користувача, у якого є варіант вибору функціоналу (безпосередньо для процесу розпізнавання або для роботи з вже розпізнаними об’єктами).

Якщо користувач обирає варіант функціоналу для виконання ідентифікації, у нього є можливість: відкрити одне зображення, відкрити каталог із зображеннями, завантажити зображення з Інтернет-ресурсів за *URL*, виконати розпізнавання об’єкта та можливість збереження результатів.

У випадку вибору користувачем функціоналу для роботи з ідентифікованими об’єктами, у нього з’являється можливість ведення пошуку, видалення, додавання чи редагування даних.

Таким чином за допомогою цієї діаграми описуються можливості системи, користувачі системи, взаємодія між користувачами і системою. Діаграму прецендентів зручно використовувати при визначенні бачення майбутньої системи для того, щоб зрозуміти хто буде користувачами системи, як ці користувачі будуть використовувати систему.

Для комфортного користування інформаційною системою наведено також блок-схему (рис. 3.2.).

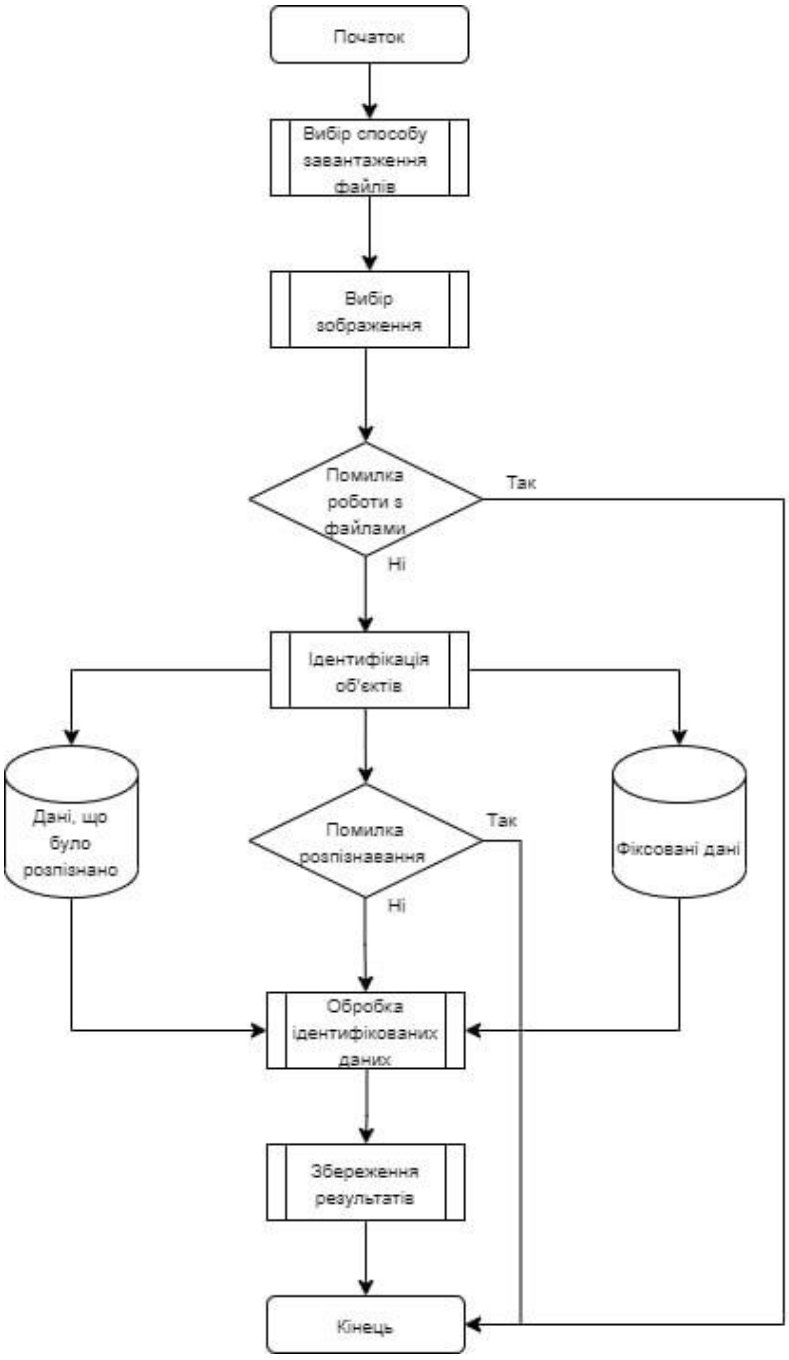

Рисунок 3.2 – Блок-схема

На початку користування інформаційною системою користувач обирає спосіб завантаження файлів зображення. Наступним кроком він обирає потрібне зображення до елементу *pictureBox*, далі виконується перевірка обраних файлів на сумісність із системою чи наявність даного файлу. Далі виконується головний функціонал розробленої системи – розпізнавання об’єктів. Результати даного процесу заносяться до Бази даних, де будуть зберігатися інформаційні дані щодо розміщення елементів на *pictureBox* та відсотку відповідності цього зображення. Також одночасно виконується перевірка на правильність виконання процесу

розпізнавання. Наступним кроком будуть виконуватися маніпуляції з даними в базі даних. Фінальним етапом роботи системи буде збереження результатів.

3.2. Вибір засобів розробки системи розпізнавання зображень

У рамках даного проекту необхідно створити інформаційну систему, що повністю підтримує процес розпізнавання та цілком вирішує поставлені задачі. Відповідно до цього розглядалися найбільш популярні та поширені способи проектування.

Тому після вивчення проблем вибору мов і технологій для розробки додатків і, виходячи з вимог до даного проекту, обрана система керування базами даних *MySQL*.

Дана система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. *MySQL* – одна з найпоширеніших систем керування базами даних. В першу чергу вона використовується для створення динамічних Веб-сторінок, оскільки має підтримку з боку різноманітних мов програмування.

MySQL вважається чудовим рішенням для додатків (малих і середніх). Вихідний код сервера накопичується та поширюється на безлічі платформ. Найповніше можливості сервера виявляються в *UNIX*-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому [8].

Для некомерційного використання *MySQL* є безкоштовним. Переваги сервера *MySQL*:

- досить простий у становленні та подальшому користуванні ;
- є можливість підтримки необмеженою кількістю користувачів, що працюють одночасно із БД;
- кількість рядків у таблицях може досягати 50 млн.;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки.

Недоліки сервера *MySQL*:

– не реалізована підтримка трансакцій, однак є можливість використання *LOCK/UNLOCK TABLE*;

– немає належної підтримки зовнішніх ключів;

– відсутня підтримка збережених процедур та тригерів;

– відсутня підтримка представлень.

Однак, перелічені недоліки не є критичними при розробці інформаційних систем для робочих груп [14].

3.3 Бібліотеки та простори імен для реалізації функціоналу

Нейронні мережеві архітектури для виявлення і розпізнавання об’єктів можна розділити на дві групи:

– Архітектури, які обробляють регіони на зображенні (*R-CNN*).

– Архітектури, які обробляють вхідне зображення цілком (YOLO, SSD).

Проаналізуємо існуючі рішення та виберемо для вирішення поставленої задачі найбільш вдалий варіант. Популярні бібліотеки в нейронних мережах для розпізнавання об’єктів:

– Архітектура *YOLO* спочатку розроблялася для задач реального часу. В алгоритмі *YOLO* зображення розділяється на осередки з використанням сітки. Для кожного осередку сітки оцінюється ймовірність присутності об’єкта взагалі, потім будуються кілька найбільш ймовірних положень об’єкта у вигляді прямокутників з центром в цій комірці, після чого для кожного отриманого прямокутника виконується оцінка ймовірностей наявності в ньому об’єктів кожного розглянутого класу.

У методі *YOLO* результати виявлення представляються у вигляді тензора розміром 7 × 7 × 1024. У разі *YOLOv3* для виділення ознак використовується згорткова нейронна мережа, яка складається з 53 шарів, як фільтри використовуються згортки розміром 3 × 3 і 1 × 1 і *Residual* блоки, які додають до виходу поточного шару значення з виходу попереднього шару. Також варто відзначити, що в *YOLOv3* виявлення об’єктів виконується на трьох масштабах,

що дозволило збільшити якість виявлення невеликих об’єктів. Мережа масштабує вхідне зображення поки не досягне першого рівня виявлення[19].

В архітектурі *YOLO* застосовують функцію *softmax* для перетворення оцінок в ймовірності класів, підсумовування яких по всіх класах дає одиницю. *YOLO* використовує класифікацію з декількома мітками. У *YOLO* функція активації *softmax* замінюється на незалежні логістичні класифікатори для обчислення ймовірності виходу, що належить певній мітці.

– Архітектура *Faster R-CNN*. Для вирішення завдання виявлення об’єктів *Faster R-CNN* в даний час є однією з найпоширеніших використовуваних архітектур на основі глибокого навчання. Попередниками даної архітектури є *R-CNN* і *Fast R-CNN*. Робота *R-CNN* складається з трьох основних етапів: початкове зображення розбивається на регіони, в яких можуть знаходитися об’єкти; далі кожен регіон подається на вхід відповідної навченої згорткової нейронної мережі, яка витягує вектор ознак для свого регіону; і згодом вектори ознак подаються на вхід набору *SVM*, виконують функцію класифікації. Кожна *SVM* навчена для визначення одного класу об’єктів.

Архітектура *R-CNN* показала високі показники точності виявлення об’єктів, але були відзначені такі недоліки, як високі витрати пам’яті й часу на навчання і обробку зображень. Тому були запропоновані модифікації архітектури, що призвели до створення *Fast R-CNN*. З точки зору швидкості метод *Fast R-CNN* має значну перевагу перед *R-CNN*, але ще одним недоліком був алгоритм вибору регіонів кандидатів. Модифікація даного етапу привела до створення *Faster R-CNN*. Архітектура *Faster R-CNN* в даний час дозволяє досягнути високої точності виявлення об’єктів і вважається відносно швидкою. При цьому збережена головна ідея вихідної архітектури *R-CNN*: виділення на зображенні регіонів, в яких можливо знаходяться об’єкти, і класифікація вмісту цих регіонів.

– Архітектура *SSD* забезпечує значний приріст швидкості обробки в порівнянні з *Faster R-CNN*. Якщо остання виконує вибір регіонів- кандидатів і класифікацію регіонів в два окремих етапи, то *SSD* виконує ці дії одночасно при обробці всього зображення. Роботу *SSD* можна описати наступним чином.

Початкове зображення проходить через ряд згортальних шарів, що в результаті дає набір карт ознак для різних масштабів. У кожній точці кожної карти ознак застосовується згортковий фільтр.

На відміну від *R-CNN*, де в регіонах-кандидатах є хоча б мінімальна ймовірність знаходження об'єкта, в *SSD* крок фільтрації регіонів відсутній. В результаті генерується набагато більша кількість описаних прямокутників на різних масштабах в порівнянні з *R-CNN*, і велика частина не містить об'єкт. З метою вирішення даної проблеми в *SSD* використовується придушення не максимумів для об'єднання схожих один на одного прямокутників в один. Вибір регіонів-кандидатів і класифікація виконуються одночасно. На останньому етапі застосовується функція *softmax* для класифікації об'єктів.

У роботі обрано бібліотеку YOLO (*You Only Look Once)* – це згорткова архітектура нейронної мережі, призначена для виявлення об'єктів. До *YOLO* класифікатори зображень використовувалися для виконання завдання виявлення об'єкта шляхом сканування всього зображення, щоб визначити місцезнаходження об'єкта. Процес сканування всього зображення починається з попередньо визначеного вікна, яке видає логічний результат, який має значення *true*, якщо вказаний об'єкт присутній в відсканованому розділі зображення, і значення *false*, якщо це не так. Після сканування всього зображення з вікном розмір вікна збільшується, що використовується для повторного сканування зображення. Методи на основі деформованих моделей частин для виявлення об'єктів *(DPM)*, використовують цю техніку, яка називається ковзним вікном.

YOLO відрізняється від інших мереж тим, що розглядає проблему виявлення зображень як проблему регресії, а не як проблему класифікації, і підтримує одну згорткову нейронну мережу для виконання всіх вищезазначених завдань. Об'єднання всіх незалежних завдань в одну мережу має наступні переваги:

– Швидкість: мережа *YOLO* надзвичайно швидка в порівнянні зі своїми попередниками, оскільки він використовує єдину мережу згортки для виявлення об'єктів. Згортка виконується для всього вхідного зображення тільки один раз, щоб отримати передбачення.

– Менше фонових помилок: *YOLO* виконує згортку всього зображення, а не його частин, завдяки чому кодує контекстну інформацію про класи і їх появу. Вона робить менше помилок при прогнозуванні фонових виправлень як об’єктів, оскільки переглядає всі зображення і причини глобально, а не локально [20].

3.4 Програмна реалізація та інструкція користувача

Для ефективного виконання поставленого завдання та реалізації головного функціоналу необхідного програмного забезпечення необхідно підключити бібліотеку розпізнавання об’єктів *YOLO v*3, яка створена на основі нейронних мереж. Файли конфігурації, які необхідні для функціонування ідентифікації об’єктів, знаходяться у вільному доступі на репозиторії *Git*. Ці файли необхідно розмістити у каталозі створеного проекту у теку *Debug*.

Як і планувалося у інформаційній системі реалізована можливість:

– розпізнавання наявних об’єктів з зображень, які були завантажені користувачем почергово;

– розпізнавання об’єктів з всіх зображень, які були завантажені у обраному каталозі;

– завантаження зображень для ідентифікації по *URL*-адресі;

– збереження результатів ідентифікації;

– одночасно із графічним розпізнаванням виконується відображення даних (координат, розмірів та типу зображення) щодо всіх ідентифікованих об’єктів на зображенні;

– одночасно з відображення графічної ідентифікації виконується фіксування даних, щодо розпізнаного об’єкту (назва об’єкту, назва файлу та точний час та дата ідентифікації), які заносяться до під’єднаної БД. Також, відносно наявних даних у БД, реалізована можливість будь-яких маніпуляцій з даними.

Реалізована інформаційна система буде мати наступний графічний інтерфейс користувача (рис. 3.3).

Рисунок 3.3 – Головна форма системи розпізнавання об’єктів

На головній формі значний обсяг займає елемент *pictureBox*, який має стан невидимості, доки до нього не завантажили зображення. На нього будуть завантажуватися зображення для проведення ідентифікації та збереження результатів.

Знизу знаходиться елемент *datagridViev*, який буде використовуватися в якості фіксування всіх даних щодо ідентифікованих об’єктів з зображення. Правіше від першого елемента *datagridViev* знаходиться другий, який буде використовуватися для відображення всіх наявних зображень у каталозі, який був обраний користувачем. У правому кутку знаходяться клавіші для керування самим процесом ідентифікації об’єктів із зображень.

Для початку використання функціоналу програмного модуля, користувачу необхідно обрати один з двох існуючих варіантів завантаження зображення до елементу *pictureBox*. Реалізована можливість завантаження зображення по одній одиниці. Для цього необхідно натиснути на клавішу “Завантажити”. Даний функціонал реалізовується шляхом відкриття елементу *openFileDialog*, реалізація якого виглядає наступним чином:

```
if(openFileDialog1.ShowDialog() ==
DialogResult.OK)
```

```
                pictureBox1.Image =
Image.FromFile(openFileDialog1.FileName);
```

В результаті відкриється діалогове вікно, де користувачу необхідно обрати існуюче зображення (рис. 3.4). В результаті чого обране зображення буде додано до елементу *pictureBox* та буде готове до виконання розпізнавання.

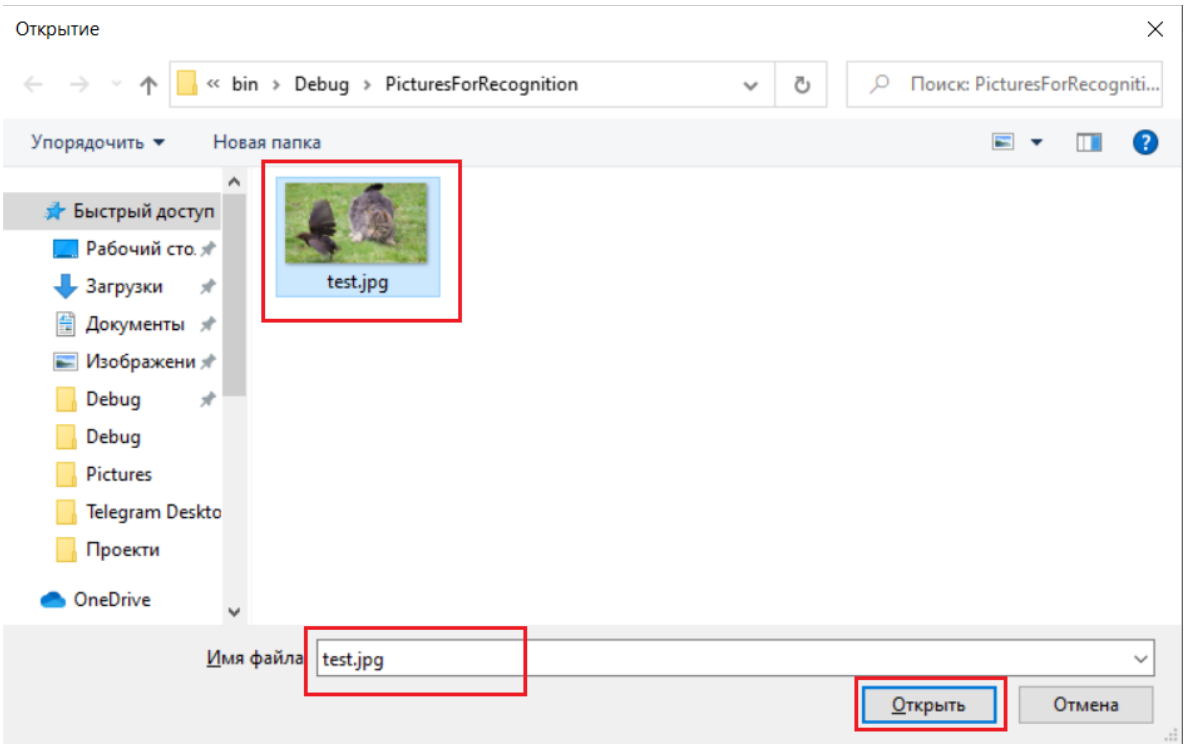

Рисунок 3.4 – Приклад одиночного зображення

Другий існуючий варіант завантаження зображень до системи – це можливість завантаження каталогу з зображеннями та додавання їх до елементу *datagridViev*. Щоб виконати завантаження необхідно натиснути на клавішу “Відкрити” та відкриється діалогове вікно (рис. 3.5), де користувачу надана можливість обрати та завантажити каталог.

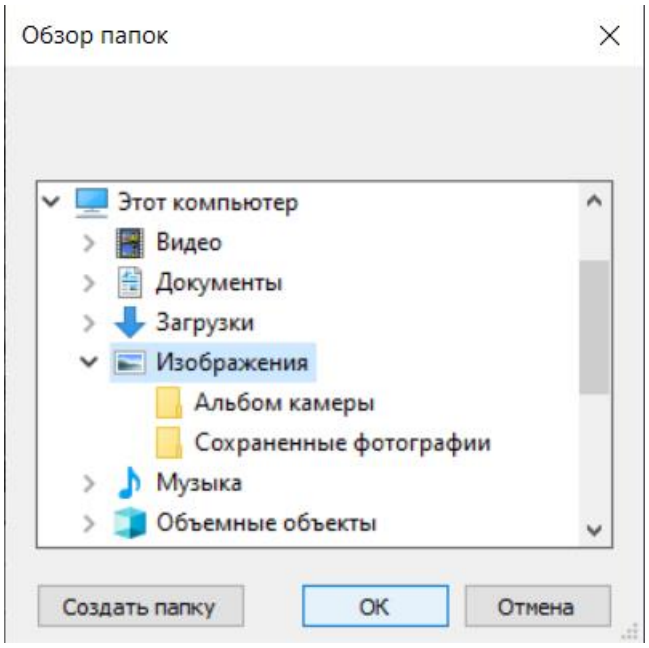

Рисунок 3.5 – Діалогове вікно для вибору необхідного каталогу

Якщо даний шлях, який обрав користувач до каталогу існує, а також містить файли форму, які підходять для зображень .За допомогою регулярних виразів, вміст каталогу буде перевірений на наявність форматів, які підходять для зображень. Реалізувати такий функціонал можна наступним чином:

```
      try
                  {
                      using (var fbd = new FolderBrowserDialog())
                      {
                          DialogResult result = fbd.ShowDialog();
                          if       (result       ==       DialogResult.OK       &&
!string.IsNullOrWhiteSpace(fbd.SelectedPath))
                          {
                              var files = Directory.GetFiles(fbd.SelectedPath);
                              var table = new DataTable();

                              table.Columns.Add("Назва файлу");
                              foreach (var file in files)
                              {
                                  if                          (Regex.IsMatch(file,
@".jpg|.png|.bmp|.jpeg|.JPG|.PNG|.BMP|.JPEG$"))
                                  {
                                      listImage.Add(file);

table.Rows.Add(Path.GetFileNameWithoutExtension(file));
                                  }
                              }
                              dataGridView2.DataSource = table;

                          }
                      }
                  }
                  catch(Exception ex)
                  {
                      MessageBox.Show(ex.Message,                          "Помилка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
```

}

При вдалому виконанні зазначених дій, користувач матиме можливість просто обираючи необхідне зображення у елементі *datagridViev*, змінювати його у елементі *pictureBox*.

Для виконання процесу ідентифікації користувачу необхідно натиснути на клавішу “Розпізнати”. Процес розпізнавання займає достатньо часу, а також потужностей ПК і прямо залежить від його характеристик. При виконанні розпізнавання створюється екземпляр класу *YoloWrapper,* у параметри якого передаються файли конфігурації: “*coco.names*”, “*yolov3.cfg*”, “*yolov3.weights*”. На основі цих файлів та зображення, яке знаходиться на елементі *pictureBox*, виконується метод *Detect()* і результати розпізнавання заносять до списку. Кожен результат знайденого об’єкту на зображенні має параметри: тип або назва об’єкту, точність розпізнавання, координати та розміри. Якщо знайдений об’єкт вдало знайдено його буде обведено в зелену рамку та зроблено відповідний червоний надпис з типом або назвою об’єкту (рис. 3.6 – 3.7).

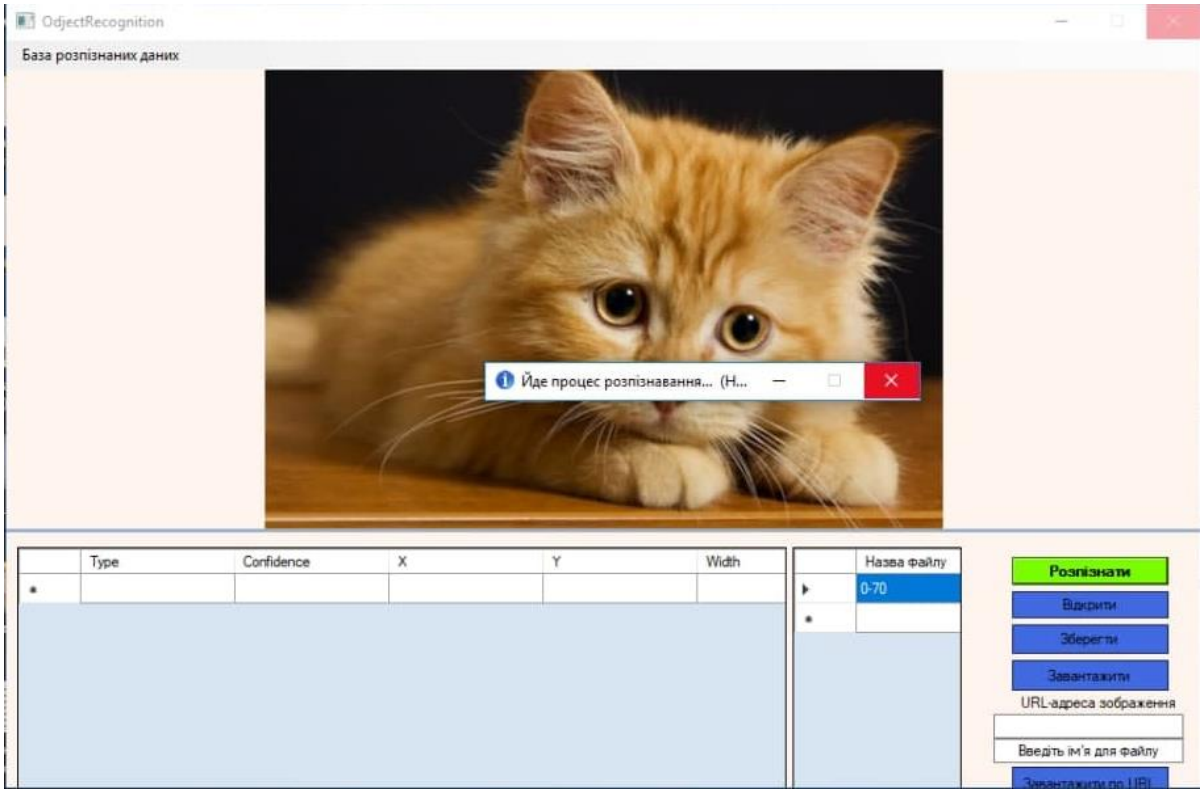

Рисунок 3.6 – Процес розпізнавання

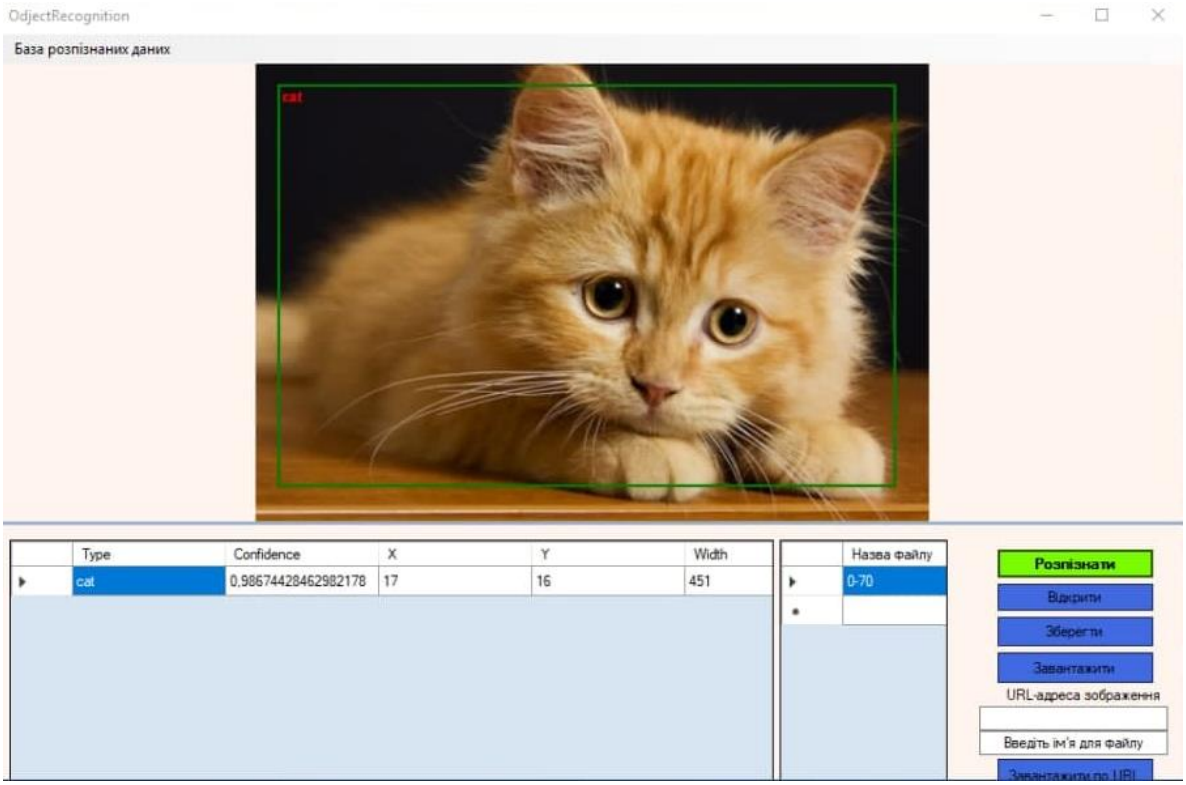

Рисунок 3.7 – Результат розпізнавання

Одночасно з відображенням графічного розпізнаванням, результати ідентифікації заносяться до елементу *datagriViev*. Користувач матиме можливість спостерігати де і який елемент та з якою точність було розпізнано.

У випадку, якщо зображення не було обране, а клавіша “Розпізнати” натиснута, користувач буде повідомлений про це належним чином. Такий процес розпізнавання реалізується таким чином:

```
      label2.Visible = true;
                              YoloWrapper        yoloWrapper        =        new
YoloWrapper("yolov3.cfg", "yolov3.weights", "coco.names");
                              MemoryStream memoryStream = new MemoryStream();
                              pictureBox1.Image.Save(memoryStream,
ImageFormat.Jpeg);

                              List<YoloItem>             yoloItems             =
yoloWrapper.Detect(memoryStream.ToArray()).ToList<YoloItem>();
                              Image image = pictureBox1.Image;
                              Graphics graphics = Graphics.FromImage(image);
                              Font   font   =   new   Font("Times   New   Roman",   9,
FontStyle.Bold);
                              SolidBrush solidBrush = new SolidBrush(Color.Red);
                              foreach (YoloItem item in yoloItems)
                              {
                                  Point point = new Point(item.X, item.Y);
                                  Size size = new Size(item.Width, item.Height);
                                  Rectangle   rectangle   =   new   Rectangle(point,
size);

                                  Pen pen = new Pen(Color.Green, 2);
                                  graphics.DrawRectangle(pen, rectangle);
                                  graphics.DrawString(item.Type,            font,
solidBrush, point);
```

```
                    dataGridView1.DataSource = yoloItems;

                    DataClass.GetData($"INSERT INTO
tb_recogobject(nameObject, nameFile, timeRecognition) VALUES ('{item.Type}',
'{Path.GetFileName(this.image)}', '{DateTime.Now}')");
                    DB.closedConn();
                }
                pictureBox1.Image = image;
                GC.Collect();
```

У розробленій системі реалізована можливість завантаження зображення із мережі Інтернет за адресою зображення. Користувачу необхідно знайти адресу необхідного зображення в мережі Інтернет, як зображено на рис. 3.8.

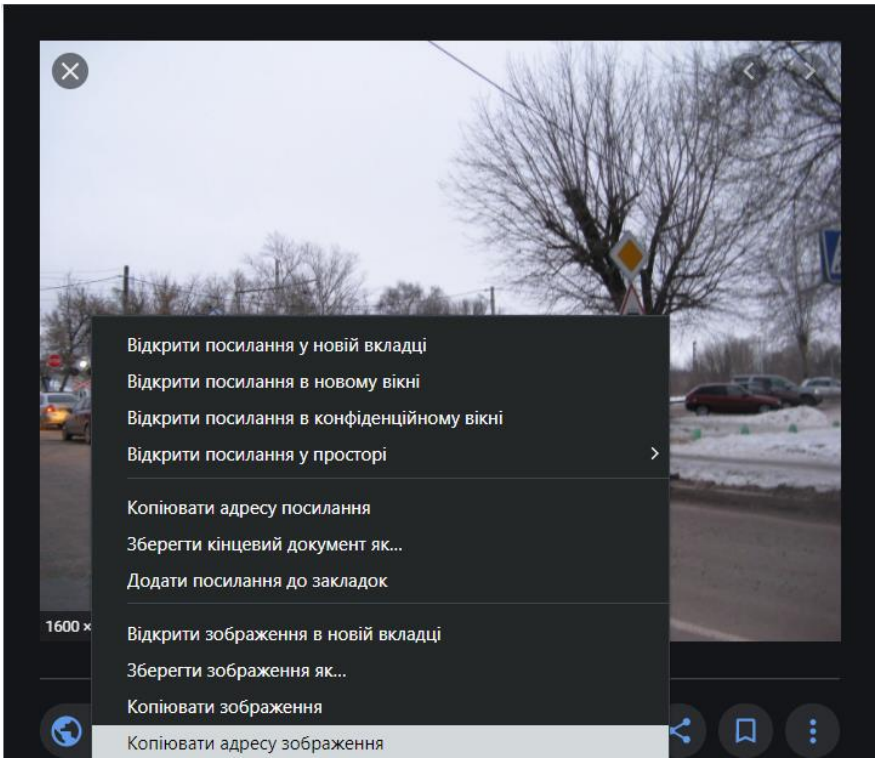

Рисунок 3.8 – Копіювання адреси необхідного зображення

Скопійовану адресу користувачу необхідно вставити до поля для *URL*-адреси, що розміщено на головному екрані системи, та ввести назву для автоматичного збереження даного файлу до теки *PicturesForRecognition.* Обраний файл з мережі Інтернет автоматично буде завантажено до елементу *pictureBox* і можливо виконувати подальшу роботу над ним.

Також одним із функціональних можливостей є можливість збереження результатів розпізнавання того зображення, яке знаходиться на елементі *pictureBox.* Для цього користувачу необхідно натиснути на клавішу “Зберегти” і у відкритому діалоговому вікні ввести назву для плануючого зображення (рис. 3.9).

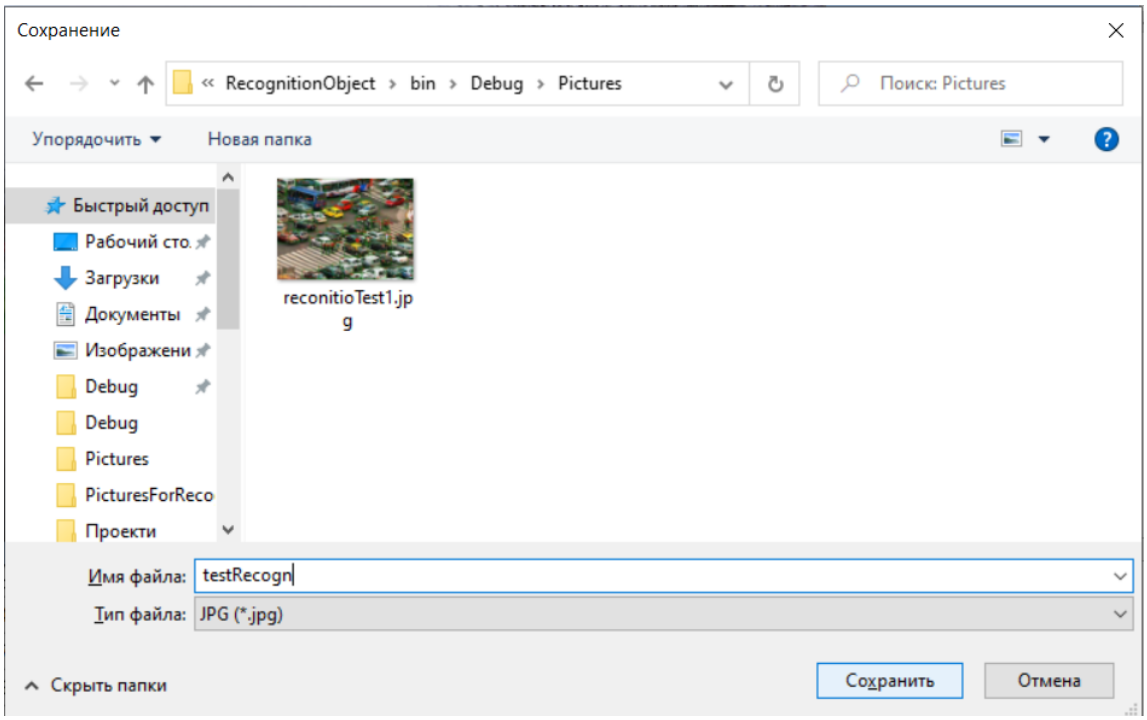

Рисунок 3.9 – Збереження результатів розпізнавання

Одночасно з розпізнаванням об’єкту із зображення виконується фіксація всіх розпізнаних об’єктів до БД. До таблиці заносяться: назва об’єкту, назва файлу, на основі якого виконувалося розпізнавання, точна дата та час виконання ідентифікації.

Для цього на розгорнутому локальному сервері *MySQL Server* за допомогою веб-додатку *phpMyAdmin* створюємо БД з назвою *db_recognation.*

У БД створюємо таблицю *tb_recognation*, де обов’язково створюється первинний ключ, який не може бути негативним, а також автоматично додається. Для кожного поля визначається тип даних, який буде там зберігатися, довжина для типу *varchar*, а також кодування *utf8_general_ci* для роботи з кирилицею. Також для кожної таблиці і полів і їх визначається тип зберігання даних *MyISAM*, який надає велику перевагу у виконанні запитів на висновок і створення нових даних (рис. 3.10).

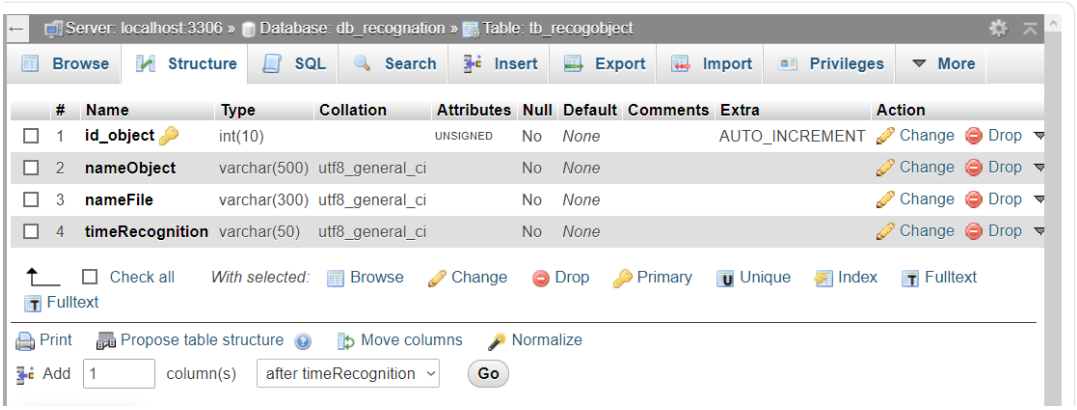

Рисунок 3.10 – Створена таблиця

Для відображення форми з результатами наявних даних та виконання маніпуляцій над даними створено елемент *menuStrip*, на якому розміщено назву “База розпізнаних даних”, для виконання переходу на дану форму.

Для ефективної реалізації інформаційної системи розроблено блок з’єднання з розробленою БД. Для цього реалізовано клас *DB.cs*, що виконує з’єднання з локальним сервером, на якому знаходиться реалізована БД (рис. 3.11).

```
1 reference
class DB
{
    static public MySqlConnection connection = new MySqlConnection("server=localhost;port=3306;username=root;password=root;database=db_recognation");

    0 references
    static public void ShowState()
    {
        MessageBox.Show(connection.State.ToString(), "Статус підключення до БД", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    0 references
    static public void openConn()
    {
        if (connection.State == System.Data.ConnectionState.Closed)
            connection.Open();
    }

    1 reference
    static public void closedConn()
    {
        if (connection.State == System.Data.ConnectionState.Open)
            connection.Close();
    }
    0 references
    public MySqlConnection GetConnection()
    {
        return connection;
    }
}
```

Рисунок 3.11 – З’єднання з сервером

На формі реалізована можливість виконувати маніпуляції з даними за допомогою широкого функціоналу, який розташований на панелі керування. Також на кожній формі знаходиться поля для введення даних, щоб здійснити додавання і редагування даних. Значний розмір форми займає *dataGridViev* – спеціальний елемент, який необхідний для відображення всіх даних в поточні

таблиці. Внизу форми знаходиться область для пошуку необхідних користувачеві даних. Приклад форми наведено на рис. 3.12.

Рисунок 3.12 – Розроблена форма для маніпуляцій з розпізнаними даними

Для додавання нових даних в таблицю користувач заповнює всі поля (при введенні не всіх даних, користувач отримає попередження і дані не будуть додані) і натискає кнопку із зображенням плюса. Щоб перевірити наявність змін в таблиці користувачеві необхідно натиснути на клавішу з зображенням кругових стрілок, і виконається оновлення даних з таблиці в *dataGridViev*. Даний функціонал може бути корисним в тих випадках, коли система не змогла ідентифікувати об’єкт, а адміністратор помітив це. Виконання *SQL*-команд (вивід, додавання, редагування та видалення) виконується за допомогою розробленого класу *DataClass.cs,* а саме реалізований метод *GetData(),* в параметри якого необхідно додати лише *SQL*-команду та передати параметри до команди.

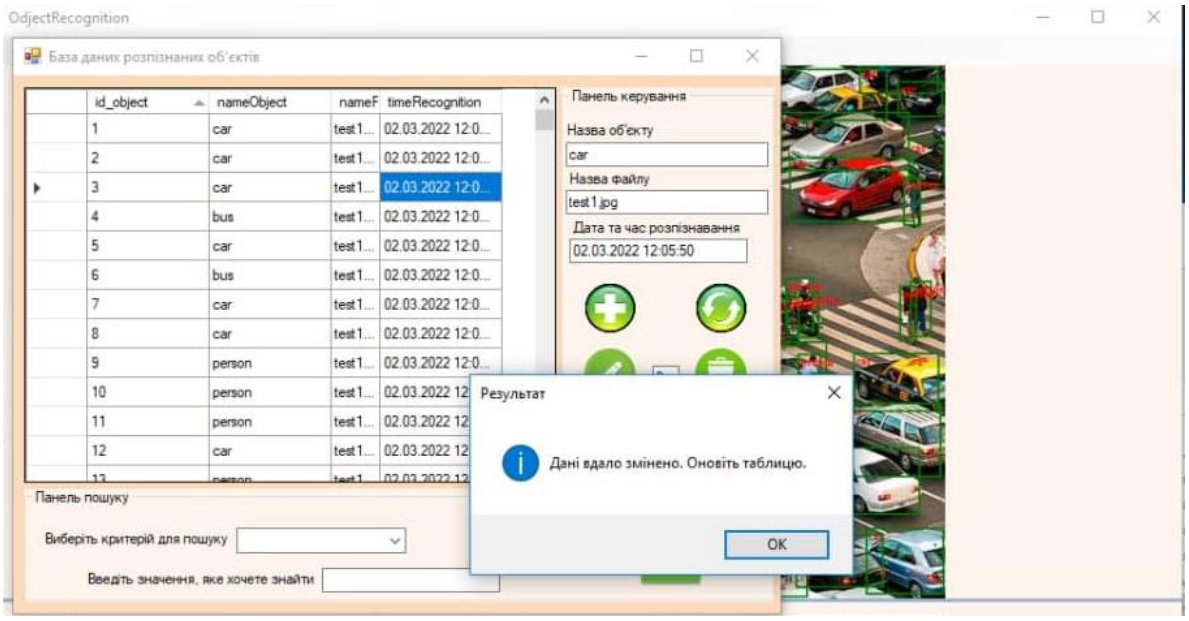

Рисунок 3.13 – Редагування даних

Для того щоб внести зміни в наявні дані таблиці, користувач заповнює всі необхідні поля, а також номер запису в поле, що знаходиться між кнопкою з зображенням олівця та кошика, і натискає кнопку із зображенням олівця. Для комфортного редагування користувачеві необхідно лише натиснути на необхідний запис в таблиці (*dataGridViev*) і необхідні записи з'являться у відповідних полях. Для видалення необхідного запису, користувачеві необхідно ввести номер і натиснути клавішу з зображенням кошика. Дані операції подано на рис. 3.13-3.15.

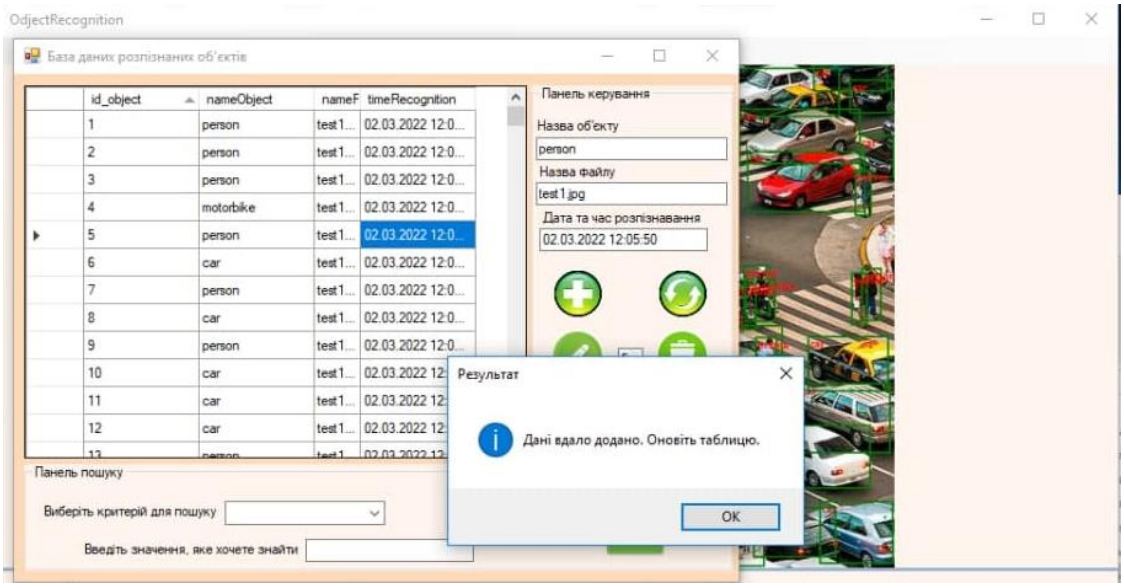

Рисунок 3.14 – Додавання нових даних

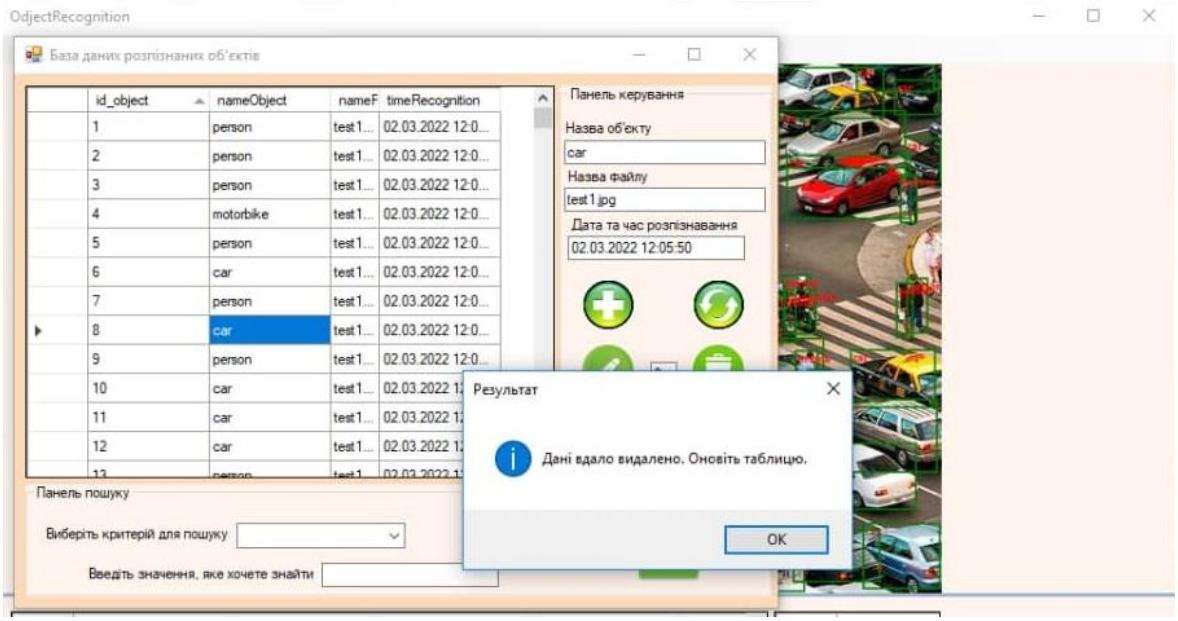

Рисунок 3.15 – Видалення даних

В розробленій інформаційній системі реалізована можливість формування файлу *excel*. Ця можливість може бути корисна для відправки звітів іншим користувачам. Для цього адміністратор системи має на активній формі натиснути на кнопку із зображенням *excel* і вибрати шлях для збереження і вказати назву майбутнього файлу (рис. 3.16).

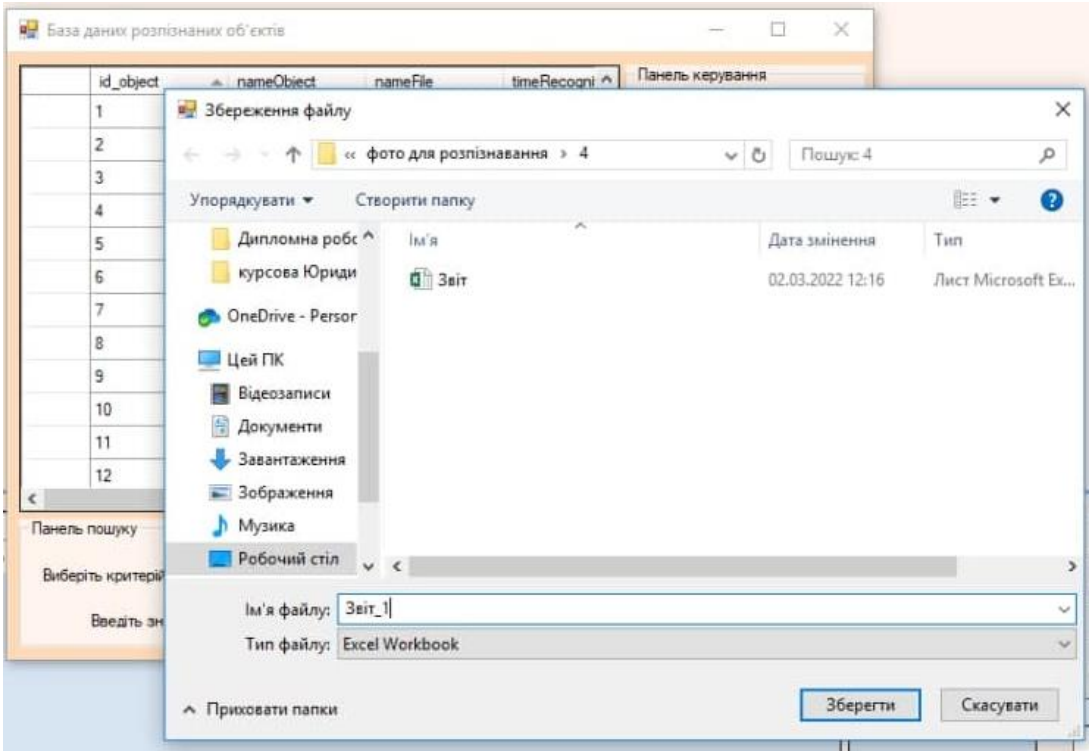

Рисунок 3.16 – Формування звіту *excel*

Для пошуку необхідних даних користувачеві необхідно на області пошуку за категоріями обрати критерії пошуку і ввести значення, за яким необхідно знайти дані. Якщо дані будуть знайдені в базі даних, то необхідні записи автоматично будуть виведені до *dataGridViev* (рис. 3.17).

Рисунок 3.17 – Пошук за категоріями

3.5. Тестування розробленої системи

Тестування реалізованої інформаційної системи здійснювалося на ПК з операційною системою *Windows 10 Pro*, обсягом оперативної пам'яті 8,00 ГБ, та на базі процесора *Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz 2.30 GHz*. Для початкової роботи запустимо файл *RecognitionObject.exe*.

Після успішного запуску програми маємо можливість завантажити та відкрити файл як один елемент, так і каталог із кількома зображеннями. Після завантаження файлу із зображенням відбувається автоматичний запуск процесу розпізнавання. Для ефективного тестування програми були використані файли різної якості та складності для розпізнавання.

Для початку перевіримо якість розпізнавання фотокартки із одним елементом (рис. 3.18).

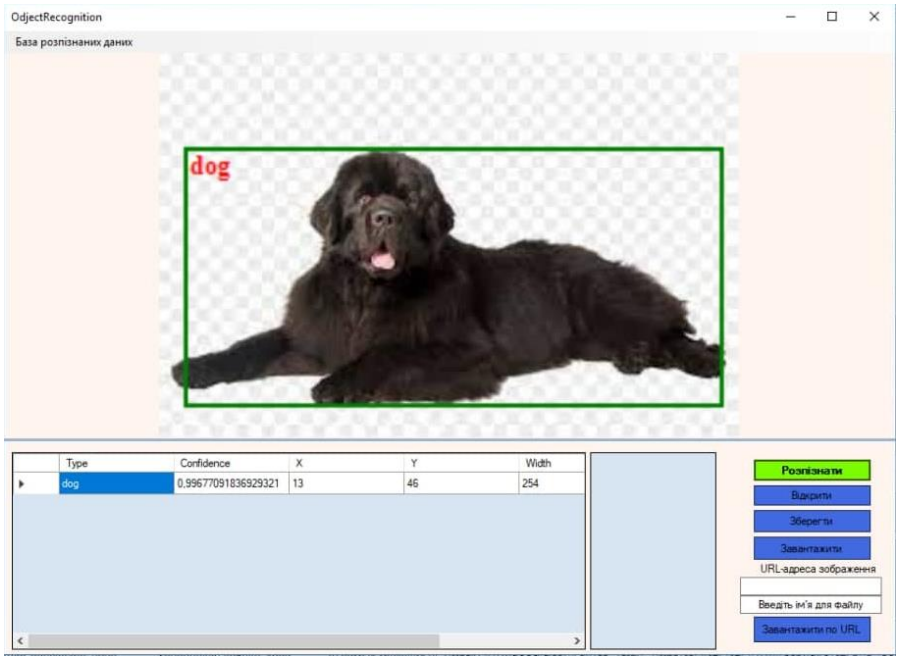

Рисунок 3.18 – Тестування зображення з одним об'єктом

Як бачимо на рисунку, програма успішно вирішила поставлену задачу. Графа *Type* ідентифікувала елемент на фотографії як *dog* із точністю 99%. На рисунку 3.19, на перший погляд, також зображений лише один елемент, однак програма ідентифікувала п'ять об'єктів, більшість із яких ледь помітна.

Центральний видимий об’єкт ідентифікований із найбільший відсотком коректності (Cat – 99%).

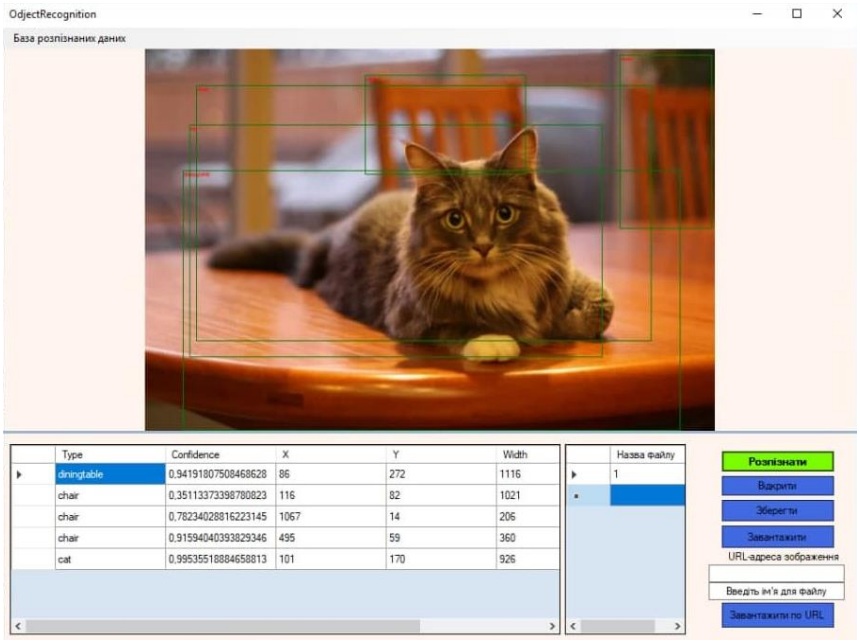

Рисунок 3.19 – Тестування зображення із кількома елементами

Далі перевіримо коректність розпізнавання із багатьма елементами на зображенні (рис 3.20).

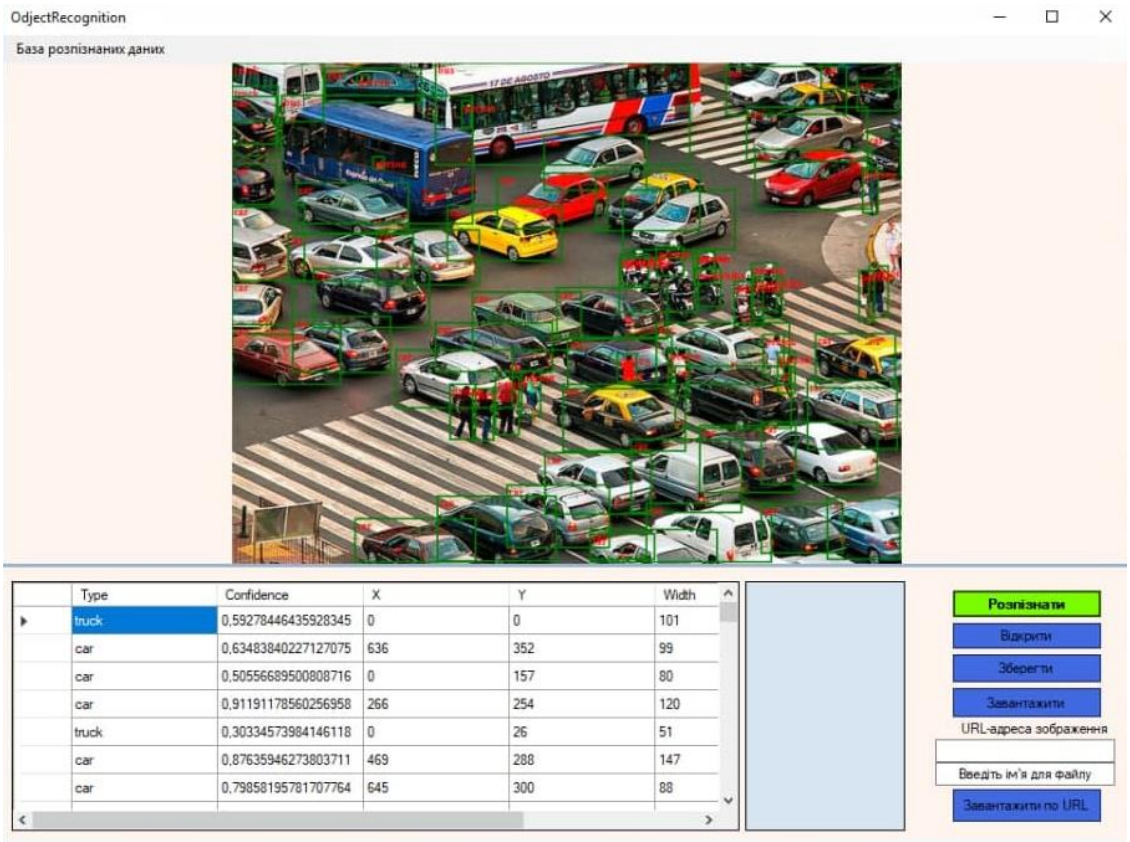

Рисунок 3.20 – Тестування зображення із багатьма елементами

Аналізуючи результати констатуємо, що програма розпізнала всі видимі елементи на фотографії незалежно від кількості елементів на фото чи їх розташування, виділивши їх та ідентифікувавши. Однак варто звернути увагу на відсоткове відношення коректності розпізнаного елементу. Предмети, що знаходяться на фото ближче до центру, чи у поний розмір, розпізнано із більшим відсотком вірогідності.

Для розробленої інформаційної системи також важливим є те, щоб об’єкти на визначеній області були не спотвореними та високої якості. Результат даного тестування представлено на рис. 3.21.

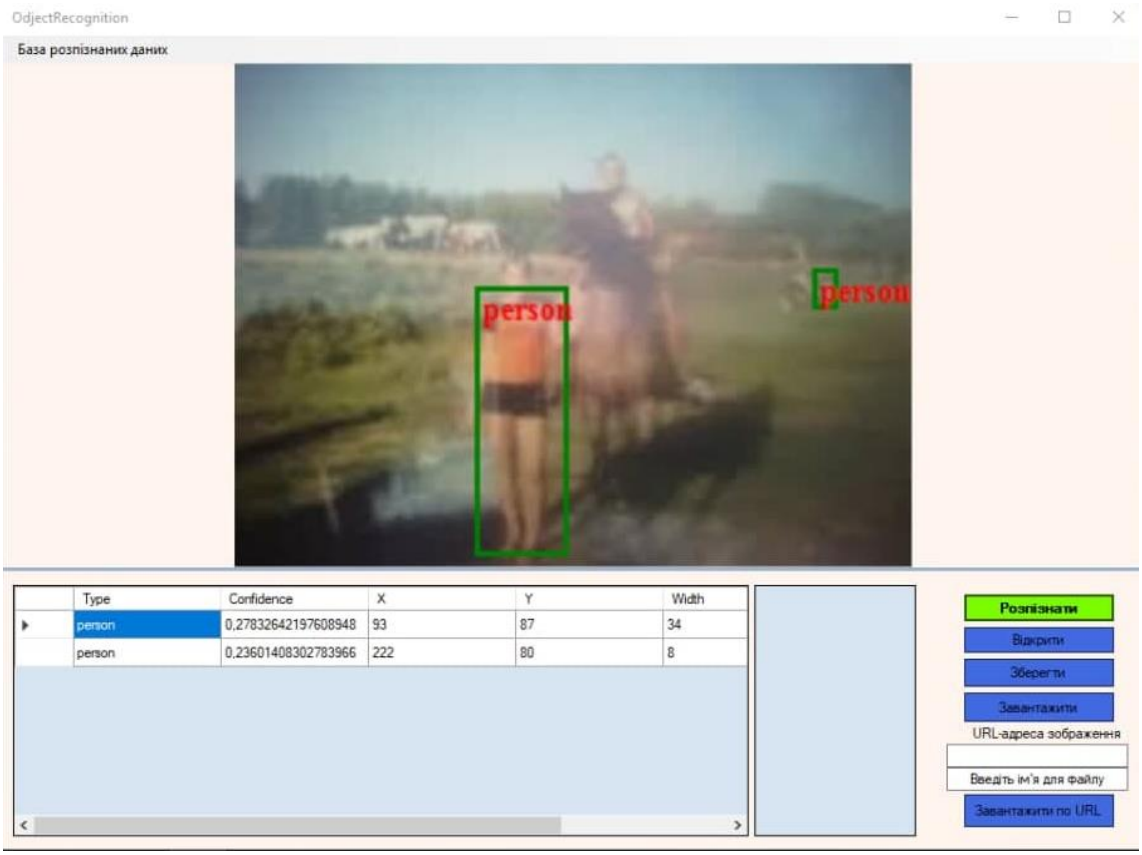

Рисунок 3.21 – Тестування зображення невисокої якості

Як бачимо, програма все-таки ідентифікувала об’єкт як Person, однак з ймовірністю лише 23-27%.

ВИСНОВКИ

У ході виконання роботи розглянуто процес розпізнавання даних та його значення на сучасному етапі розвитку, комплексно проаналізовано методи та підходи, які використовуються у технологіях розпізнавання образів на зображеннях, окреслена проблематика розпізнавання об’єктів та визначений метод її рішення на основі практики відомих досліджень. На основі здійсненого аналізу розроблено критерії, які застосовано для проведення порівняльного аналізу методів розпізнавання елементів на площині зображення.

Особливу увагу приділено сучасним архітектурним рішенням для вирішення поставленої задачі та особливостям нейронних мереж, що застосовуються для розпізнавання об’єктів на фотографії.

Розглянуто методику проведення процесу розпізнавання об’єктів, розглянуто модель та принцип побудови й роботи нейронних мереж. У роботі приділено увагу розгляду сучасних архітектурних рішень для виконання задачі та вирішено використовувати бібліотеку *YOLO*, зважаючи на її вагомі переваги.

Результатом роботи стала розроблена модернізована модель, на основі базової архітектури, нейронної мережі для розпізнавання об’єкта на зображеннях. Проведено дослідження розробленого методу, реалізованого у вигляді інформаційної системи, показана ефективність розробленого методу ідентифікації об’єктів на зображеннях. Результати експериментальних досліджень, наведені в роботі, засвідчують про ефективне використання нейронної мережі для задачі розпізнавання об’єктів.

На підставі отриманих результатів можна зробити висновок про те, що обрана інформаційна модель штучної нейронної мережі відповідає моделі реальної системи і враховує всі її характеристики. Всі поставлені завдання виконані в повному обсязі, отже, головна мета досягнута.

Матеріал, представлений у роботі, може бути з успіхом використаний у навчальному процесі для викладання дисциплін “Розподілена обробка інформації”, “Технологія проектування комп’ютерних систем”, “Середовища візуального проектування”.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lades A., Vorbruggen J., Buhmann J., Distortion invariant object recognition in the dynamic link architecture // IEEE Transactions on computers, 1993, vol. 42, no. 3, PP. 300 –310, March 1993.

2. Adini Y., Moses Y., Ullman S. Face Recognition: The Problem of Compensating for Changes in Illumination Direction // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1997. – Vol. 19. – P. 721–732.

3. Aizenberg I. N., Aizenberg N. N., Krivosheev G.A. Multi–valued and Universal Binary Neurons: Learning Algorithms, Applications to Image Processing and Recognition // Lecture Notes in Artificial Intelligence – Machine Learning and Data Mining in Pattern Recognition. – 1999. – P. 21–35.

4. Brunelli R., Poggio T. Caricatural Effects in Automated Face Perception. Massachusetts Institute of Technology, Cambridge. – 6 p.

5. Bryliuk D., Starovoitov V. Application of Recirculation Neural Network and Principal Component Analysis for Face Recognition // The 2nd International Conference on Neural Networks and Artificial Intelligence. – Minsk: BSUIR, 2001. – P.136–142.

6. Huang G., Liu Z., Weinberger K. Q. Densely connected convolutional networks, arXiv preprint arXiv:1608.06993, Aug. 2016. [Online]. Available: https://arxiv.org/abs/1608.06993v1

7. Grother P. Face Recognition Vendor Test (FRVT). Performance of Face Identification Algorithms. / Patrick Grother, Mei Ngan. – Information Access Division National Institute of Standards and Technology. – May 26, 2014 – p. 138.

8. Ortigosa–Hernández J., Inza I., Lozano J. A., Towards competitive classifiers for unbalanced classification problems: A study on the performance scores, arXiv preprint arXiv:1608.08984, Aug. 2016. [Online]. Available: https://arxiv.org/abs/1608.08984

9. Joo Er Meng, W.Chen, Wu Shiqian, “High–speed face recognition based on discrete cosine transform and RBF neural networks”, IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 679 – 691,2005.

10. M. Abadi, A. Agarwal et al., “Tensorflow: Large–scale machine learning on heterogeneous distributed systems,” arXiv preprint arXiv:1603.04467, Mar. 2016. [Online]. Available: https://arxiv.org/abs/1603.04467

11. M. Hardt and T. Ma, “Identity matters in deep learning,” arXiv preprint arXiv:1611.04231, Nov. 2016. [Online]. Available: https: //arxiv.org/abs/1611.04231

12. M. Thoma, “The HASYv2 dataset,” arXiv preprint arXiv:1701.08380, Jan. 2017. [Online]. Available: https://arxiv.org/abs/1701.08380 86

13. N. McLaughlin, J. M. D. Rincon, and P. Miller, “Dataaugmentation for reducing dataset bias in person re–identification,” in International Conference on Advanced Video and Signal Based Surveillance (AVSS), no. 12, Aug. 2015, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/ 7301739/ 24.

14. Pakhomova V. M. Network Traffic Forcasting in information-telecommunication System of Prydniprovsk Railways Based on Neuro-fuzzy Network. Science and Transport Progress. 2016. № 6 (66). pp.105–114.

15. S. Chetlur, C. Woolley et al., “cuDNN: Efficient primitives for deep learning,” arXiv preprint arXiv:1410.0759, Oct. 2014. [Online]. Available: https://arxiv.org/abs/1410.0759

16. Kobayashi K. sprocket : Open-Source Voice Conversion Software. Kobayashi K., Toda T. – Proc. Odyssey. –2018. – P. 203–210.

17. Nagrani A. VoxCeleb: a large-scale speaker identification dataset. [Text] / A. Nagrani, J. S. Chung, A. Zisserman // Proceedings of INTERSPEECH, 2017. – P. 2616–2620.

18. Y.Taigman, M.Yang, M.Ranzato, “DeepFace: Closing the Gap to Human–Level Performance in Face Verification”, [Online]. Available at: https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf

19. YOLO v1: Part 1 [Електронний ресурс]. URL: https://medium.com/adventures–with–deep–learning/yolo–v1–part–1–cfb47135f81f/.

20. YOLO: Real–Time Object Detection [Електронний ресурс]. URL: https://pjreddie.com/darknet/yolo/.

21. А. М. Лисенко Застосування біометричних систем для ідентифікації особи. Вісник Київського нац. ун.–ту ім. Т.Шевченка, Юридичні науки, 2004, №60/62, с. 87–91.

22. Васенко, Д. В. Методи навчання штучних нейронних мереж. // Інформатизація освіти. – 2017. – С. 20–29.

23. Гонсалес Р., Вудс Р. Цифрова обробка зображень. М.:Техносфера, 2005. 1072 с.

24. Гудфеллоу Я., Бенджио И., Курвилль А. Глибоке навчання – Deep Learning. М.: ДМК Пресс, 2017. 652 с.

25. Ечеагарай-Патрон Б. А., Кобер В. І. Метод розпізнавання осіб з використанням тривимірних поверхонь. // Інформаційні процеси. – 2016. – Т. 16. № 2. – С. 170-176.

26. Желтов, С. Ю. Обробка та аналіз зображень в задачах машинного зору. М .: Фізматкніга, 2010. 672 с.

27. Курочкина И.П., Калинин И.И., Маматова Л.А., Шувалова Е.Б. Нейронні моделі в діагностиці фінансового результату підприємств житлово-комунального господарства. Статистика і математичні методи в экономіці. 2019. № 3. С. 52–60.

28. Мозолевська М. О. Використання нейронних мереж для прогнозування у фінансовій сфері [Електронний ресурс] / Мозолевська М. О., Ставицький О. В. // Актуальні проблеми економіки та управління: збірник наукових праць молодих вчених. – Електронні текстові дані (1 файл: 443 Кбайт). – 2017. – Вип. 11. – Режим доступу: https://ela.kpi.ua/handle/123456789/22609 .

29. Максименко О. А. Генерація цільового голосу людини з використанням нейронних мереж. – К.:НТУУ «КПІ імені Ігоря Сікорського», 2019. – 62 с.

30. Ніколенко С. Кадурін А., Архангельська Е. Глибоке навчання. Занурення в світ нейронних мереж, 2018. 480 с.

31. Пахомова В. М. Теорія проектування комп’ютерних[Текст]. Методичні вказівки до виконання практичних робіт. – Дніпро: ДІІТ, 2017. – 207 с.

32. Принципи ущільнення та перетворення зображення : монографія / за заг. ред.В. П. Кожем'яко. – Вінниця : ВНТУ, 2011. – 242 с.

33. Рауменко В.А. Застосування нейронних мереж для вирішення практичних задач в економіці. Вектор економіки. 2019. № 10. С. 1–12.

34. Тарасенко–Клятченко О.В., Буц В.В. Організація багатоетапного методу навчання згорткової нейронної мережі. // Міжнародний науковий журнал «Інтернаука» 2018. Випуск 6. – С. 44-46. URL: https://www.inter–nauka.com/issues/2018/6/3624/

35. Теслюк В. Основні компоненти нейромереж з паралельною обробкою даних вертикальної групи в режимі реального часу. Advances in Intelligent Systems and Computing II, Advances in Intelligent Systems and Computing, Springer International Publishing AG, 2018. с.558–576.

36. Флах П. Машинне навчання. Наука і мистецтво побудови алгоритмів, які вилучають знання із даних. – М.: ДМК Пресс, 2015. 400 с.

37. Цмоць І. С. Основні вертикально-паралельні компоненти нейронної мережі реального часу. Матеріали XII Міжнародної науково-технічної конференції CSIT 2017, 5–8 вересня 2017 р. Львів, Україна, 344–347

38. Шевченко И.В., Шкарупа Н.С., Гончар М.В. Модель и метод побудови багаторівневої системи розпізнавання. // Вісник Кременчуцького національного університету імені Михайла Остроградського. Кременчук: КрНУ, 2017. Випуск 6. – С. 54–63.

39. Яковлєв Ю.В. Про оцінку ефективності застосування FPGA у складі PIM-систем. // Системи керування та машини. – 2016. – № 1. – С.56-61.

ДОДАТКИ

ДОДАТОК А

```
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Drawing;
using System.Drawing.Imaging;
using System.Linq;
using System.Windows.Forms;
using Alturos.Yolo;
using Alturos.Yolo.Model;
using System.Text.RegularExpressions;
using System.Net;

namespace RecognitionObject
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {

      if(textBox1.Text != "" && textBox2.Text != "Введіть ім'я для файлу")
      {
        WebClient client = new WebClient();
        Uri uri = new Uri(textBox1.Text);

          string filename = textBox2.Text;
        client.DownloadFileAsync(uri, Application.StartupPath + $"\\PicturesForRecognition\\{filename}.jpg");
      }
    }

    private void button2_Click(object sender, EventArgs e)
    {
      try
      {
        YoloWrapper yoloWrapper = new YoloWrapper("yolov3.cfg", "yolov3.weights", "coco.names");
        MemoryStream memoryStream = new MemoryStream();


        pictureBox1.Image.Save(memoryStream, ImageFormat.Jpeg);

        List<YoloItem> yoloItems = yoloWrapper.Detect(memoryStream.ToArray()).ToList<YoloItem>();

        Image image = pictureBox1.Image;

        Graphics graphics = Graphics.FromImage(image);

        Font font = new Font("Times New Roman", 9, FontStyle.Bold);
        SolidBrush solidBrush = new SolidBrush(Color.Red);
        foreach (YoloItem item in yoloItems)
        {
          Point point = new Point(item.X, item.Y);
          Size size = new Size(item.Width, item.Height);
          Rectangle rectangle = new Rectangle(point, size);
          Pen pen = new Pen(Color.Green, 2);
```

```
                graphics.DrawRectangle(pen, rectangle);
                graphics.DrawString(item.Type, font, solidBrush, point);
                dataGridView1.DataSource = yoloItems;

                DataClass.GetData($"INSERT INTO `tb_recogobject`(`nameObject`, `nameFile`, `timeRecognition`) VALUES
('{item.Type}', '{Path.GetFileName(this.image)}', '{DateTime.Now}')");
                DB.closedConn();
            }

            pictureBox1.Image = image;
            GC.Collect();
        }
        catch(Exception ex)
        {
            MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

    }

    private void базаРозпізнанихДаних ToolStripMenuItem_Click(object sender, EventArgs e)
    {
        infoObject infoObject = new infoObject();
        infoObject.Show();
    }
    List<string> listImage;
    string image;
    private void button4_Click(object sender, EventArgs e)
    {
        try
        {
            using (var fbd = new FolderBrowserDialog())
            {
                DialogResult result = fbd.ShowDialog();
                if (result == DialogResult.OK && !string.IsNullOrWhiteSpace(fbd.SelectedPath))
                {
                    var files = Directory.GetFiles(fbd.SelectedPath);
                    var table = new DataTable();

                    table.Columns.Add("Назва файлу");
                    foreach (var file in files)
                    {
                        if (Regex.IsMatch(file, @".jpg|.png|.bmp|.jpeg|.JPG|.PNG|.BMP|.JPEG$"))
                        {
                            listImage.Add(file);
                            table.Rows.Add(Path.GetFileNameWithoutExtension(file));
                        }
                    }
                    dataGridView2.DataSource = table;
                }
            }
        }
        catch(Exception ex)
        {
            MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

    }

    private void Form1_Load(object sender, EventArgs e)
    {
        listImage = new List<string>();
    }

    private void dataGridView2_SelectionChanged(object sender, EventArgs e)
    {
        this.image = listImage[dataGridView2.CurrentRow.Index];
        var image = new Bitmap(this.image);
        pictureBox1.Image = image;
```

```
        }

        private void button3_Click(object sender, EventArgs e)
        {
            try
            {
                Bitmap bmpSave = (Bitmap)pictureBox1.Image;
                SaveFileDialog sfd = new SaveFileDialog();
                sfd.DefaultExt = "bmp";
                sfd.Filter = "JPG|*.jpg|PNG|*.png|BMP*.bmp|";
                if (sfd.ShowDialog() == DialogResult.OK)

                    bmpSave.Save(sfd.FileName, ImageFormat.Jpeg);
            }
            catch(Exception ex)
            {
                MessageBox.Show(ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }


        }

        private void button5_Click(object sender, EventArgs e)
        {
            if(openFileDialog1.ShowDialog() == DialogResult.OK)
             pictureBox1.Image = Image.FromFile(openFileDialog1.FileName);
        }
    }
}
```

ДОДАТОК Б

```
using System;
using System.Net;
using System.Windows.Forms;
using Excel = Microsoft.Office.Interop.Excel;


namespace RecognitionObject
{
   public partial class infoObject : Form
   {
      public infoObject()
      {
         InitializeComponent();
         dataGridView1.DataSource = DataClass.GetData("SELECT * FROM `tb_recogobject`");
      }

      private void pictureBox1_Click(object sender, EventArgs e)
      {
         if (txtName.Text == "" || txtFile.Text == "" || txtDate.Text == "")
         {
            MessageBox.Show("Введено пропущені дані!", "Помилка", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
            return;
         }
         else
         {
            try
            {
               DataClass.GetData($"INSERT INTO `tb_recogobject`(`nameObject`, `nameFile`, `timeRecognition`) VALUES
('{txtName.Text}', '{txtFile.Text}', '{txtDate.Text}')");
               MessageBox.Show("Дані вдало додано. Оновіть таблицю.", "Результат", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
            catch (Exception ex)
            {
               MessageBox.Show(ex.Message, "Помилка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }

            txtFile.Text = "";
            txtDate.Text = "";
            txtName.Text = "";
         }
      }

      private void pictureBox2_Click(object sender, EventArgs e)
      {
         dataGridView1.DataSource = DataClass.GetData("SELECT * FROM `tb_recogobject`");
      }

      private void pictureBox3_Click(object sender, EventArgs e)
      {
         DataClass.GetData($"UPDATE `tb_recogobject` SET
`nameObject`='{txtName.Text}',`nameFile`='{txtFile.Text}',`timeRecognition`='{txtDate.Text}' WHERE
`id_object`={textBox4.Text}");
         MessageBox.Show("Дані вдало змінено. Оновіть таблицю.", "Результат", MessageBoxButtons.OK,
MessageBoxIcon.Information);
         txtFile.Text = "";
         txtDate.Text = "";
         txtName.Text = "";
         textBox4.Text = "";
      }

      private void pictureBox4_Click(object sender, EventArgs e)
      {
```

```
            DataClass.GetData($"DELETE FROM `tb_recogobject` WHERE `id_object`={textBox4.Text}");
            MessageBox.Show("Дані вдало видалено. Оновіть таблицю.", "Результат", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            textBox4.Text = "";
        }

        private void pictureBox5_Click(object sender, EventArgs e)
        {
            SaveFileDialog saveFileDialog = new SaveFileDialog() { Filter = "Excel Workbook|*.xlsx" };
            saveFileDialog.ShowDialog();
            string path = saveFileDialog.FileName;
            Excel.Application application = new Excel.Application();
            Excel.Workbook workbook = application.Workbooks.Add();
            Excel.Worksheet worksheet = workbook.ActiveSheet;


            for (int i = 1; i < dataGridView1.RowCount + 1; i++)
            {
                for (int j = 1; j < dataGridView1.ColumnCount + 1; j++)
                {
                    worksheet.Rows[i].Columns[j] = dataGridView1.Rows[i - 1].Cells[j - 1].Value;
                }
            }
            application.AlertBeforeOverwriting = false;
            workbook.SaveAs(path);
            application.Quit();
        }

        private void pictureBox6_Click(object sender, EventArgs e)
        {
            textBox4.Text = "";
            txtFile.Text = "";
            txtDate.Text = "";
            txtName.Text = "";
        }

        private void pictureBox7_Click(object sender, EventArgs e)
        {
            if(comboBox1.SelectedIndex == 0) dataGridView1.DataSource = DataClass.GetData($"SELECT * FROM
`tb_recogobject` WHERE `nameObject` = '{textBox1.Text}'");
            else if(comboBox1.SelectedIndex == 1) dataGridView1.DataSource = DataClass.GetData($"SELECT * FROM
`tb_recogobject` WHERE `nameFile` = '{textBox1.Text}'");
            else if (comboBox1.SelectedIndex == 2) dataGridView1.DataSource = DataClass.GetData($"SELECT * FROM
`tb_recogobject` WHERE `timeRecognition` = '{textBox1.Text}'");
        }

        private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
        {
            textBox4.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();
            txtName.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
            txtFile.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
            txtDate.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
        }
    }
}
```

ДОДАТОК В

```
class DB
   {
      static public MySqlConnection connection = new MySqlConnection("server=localhost;port=3306;username=root;password=root;database=db_recognation");

      static public void ShowState()
      {
         MessageBox.Show(connection.State.ToString(), "Статус підключення до БД", MessageBoxButtons.OK, MessageBoxIcon.Information);
      }

      static public void openConn()
      {
         if (connection.State == System.Data.ConnectionState.Closed)
            connection.Open();
      }

      static public void closedConn()
      {
         if (connection.State == System.Data.ConnectionState.Open)
            connection.Close();
      }
      public MySqlConnection GetConnection()
      {
         return connection;
      }
   }
```

ДОДАТОК Г

```
class DataClass
   {
      static public DataTable GetData(string qwery)
      {
         DataTable dataTable = new DataTable();
         MySqlConnectionStringBuilder mysqlCSB;
         mysqlCSB = new MySqlConnectionStringBuilder();
         mysqlCSB.Server = "localhost";
         mysqlCSB.Database = "db_recognation";
         mysqlCSB.UserID = "root";
         mysqlCSB.Password = "root";



         using (MySqlConnection con = new MySqlConnection())
         {
            con.ConnectionString = mysqlCSB.ConnectionString;
            MySqlCommand com = new MySqlCommand(qwery, con);
            try
            {
               con.Open();
               using (MySqlDataReader dr = com.ExecuteReader())
               {
                  if (dr.HasRows)
                  {
                     dataTable.Load(dr);
                  }
               }
            }

            catch (Exception ex)
            {
               MessageBox.Show(ex.Message);
            }
         }
         return dataTable;


      }
   }
```

ДОДАТОК Д

```
static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
```