

РІВНЕНСЬКИЙ ДЕРЖАВНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ

Факультет математики та інформатики

Кафедра інформаційних технологій та моделювання

Кваліфікаційна робота

за освітнім ступенем «магістр»

на тему:

**ІДЕНТИФІКАЦІЇ ГРАФІЧНИХ ОБ'ЄКТІВ ДЛЯ ЗАСТОСУВАННЯ В
СТОМАТОЛОГІЇ**

Виконав:

здобувач 2 курсу

групи М-КН-21

спеціальності 122 «Комп'ютерні науки»

Бойко Олег Борисович

Науковий керівник:

к. п. н., доцент Петренко С. В.

Рівне – 2024

ЗМІСТ

РОЗДІЛ 1. РОЗПІЗНАВАННЯ ОБРАЗІВ	8
1.1. Предмет розпізнавання образів	8
1.2. Історія формування теорії розпізнавання образів	8
1.3. Проблема розпізнавання образів	9
1.4. Класи та їх властивості.....	10
1.5. Методи розпізнавання образів.....	15
1.6. Практичне застосування.....	18
1.7. Порівняння з аналогами	19
Висновки до першого розділу.....	20
РОЗДІЛ 2. ОБРОБКА ЗОБРАЖЕНЬ. БІБЛІОТЕКА EMGU.CV	23
2.1. Обробка зображень	23
2.2. Історія	24
2.3. Види комп'ютерної графіки.....	25
2.4. Основні методи обробки зображень	27
2.5. EMGUCV	28
2.5.1. Історія EMGUCV	30
2.5.2. Застосування	31
Висновки до другого розділу	33
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ЗАСТОСУНКУ	36
3.1. Технологія WPF.....	37
3.1.1. Переваги WPF	38
3.1.2. Архітектура WPF	40
3.2. Технологія JSON	42
3.3. Інсталяція MS Visual Studio і його налаштування	44
3.4. Початок роботи та ознайомлення з інтерфейсом	46
3.4.1. Під'єднання бібліотеки EMGUCV	48
3.5. Алгоритм попередньої обробки зображення	48
Висновки до третього розділу	51
РОЗДІЛ 4. ІНСТРУКЦІЯ КОРИСТУВАЧА	53
Висновки до четвертого розділу.....	56

ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60

ВСТУП

У сучасній стоматології все більшого значення набуває використання інформаційних технологій для автоматизації діагностичних і лікувальних процесів. Одним із ключових напрямків є розробка методів ідентифікації графічних об'єктів, таких як рентгенівські знімки, панорамні зображення зубів, 3D-моделі щелепи та інші візуальні дані.

Ця робота спрямована на створення програмного забезпечення, яке дозволить ідентифікувати графічні об'єкти в стоматології, автоматизувати аналіз зображень і прискорити діагностичні процедури.

Актуальність

У стоматології часто використовуються графічні дані для постановки діагнозу, моніторингу стану пацієнта та планування лікування. Однак обробка таких даних вручну займає багато часу та залежить від суб'єктивного досвіду лікаря.

Автоматизовані системи розпізнавання графічних об'єктів дозволяють:

- підвищити точність і надійність діагностичних висновків;
- оптимізувати процес аналізу візуальних даних;
- скоротити час, необхідний для обробки інформації;
- інтегрувати дані у централізовану базу для тривалого моніторингу стану пацієнта.

Впровадження таких систем є актуальним як для великих стоматологічних центрів, так і для приватних клінік.

Об'єкт дослідження: графічні зображення, що використовуються у стоматологічній діагностиці (рентгенограми, ортопантомограми, 3D-моделі).

Предмет. Методи і алгоритми автоматизованої ідентифікації графічних об'єктів для застосування у стоматології.

Мета дослідження

Розробити програмний засіб для ідентифікації графічних об'єктів, що дозволить автоматизувати аналіз стоматологічних зображень.

Завдання. Для досягнення поставлених цілей потрібно:

- Дослідити предметну область стоматології для визначення вимог до програмного забезпечення.
- Вивчити сучасні методи обробки графічних зображень і алгоритми ідентифікації.
- Розробити програмний інструмент для автоматизованої ідентифікації стоматологічних графічних об'єктів із використанням бібліотеки EMGUCV.
- Реалізувати систему із зручним інтерфейсом, який забезпечить обробку даних у режимі реального часу.
- Протестувати програмний виріб та провести аналіз його ефективності.

Гіпотеза. Використання автоматизованого інструменту для ідентифікації стоматологічних графічних об'єктів дозволить:

- зменшити час обробки зображень;
- підвищити точність діагностики;
- забезпечити тривалий моніторинг змін стану пацієнта;
- зменшити залежність діагнозу від людського фактору.

Методологія

У роботі використовується бібліотека EMGUCV, яка є обгорткою для OpenCV, і надає розширені можливості для обробки графічних об'єктів, таких як:

- сегментація зображень;
- фільтрація шумів;
- розпізнавання контурів і структур;
- ідентифікація об'єктів за заздалегідь створеними шаблонами.

Для створення програмного засобу використовується середовище Visual Studio .NET, що забезпечує можливість інтеграції з іншими медичними системами.

Очікувані результати

Створення програмного засобу, який автоматизує ідентифікацію стоматологічних графічних об'єктів.

Розробка бази даних, яка зберігатиме зображення, аналіз і результати обробки.

Зручний інтерфейс, що дозволить лікарям швидко обробляти інформацію та зменшити час на постановку діагнозу.

Надійність і вірогідність результатів забезпечено обґрунтованістю вихідних методологічних та теоретичних положень дослідження, використанням валідних методів і методик дослідження, застосуванням методів статистичного аналізу отриманих даних, репрезентативністю вибірки.

Практичне значення одержаних результатів полягає в можливості використання розробленого програмного рішення в реальній клінічній практиці стоматологічних установ. Автоматизоване і точне оброблення зображень дозволяє зменшити час на виконання рутинних завдань, покращити якість діагностики і прийняття рішень, а також забезпечити підвищену точність і ефективність лікування пацієнтів. Розробка також має значний потенціал для підвищення рівня безпеки та стандартизації процедур, зокрема, у використанні зображень для віртуальних консультацій і дистанційного моніторингу стану пацієнтів.

Апробація результатів дослідження. Основні результати дослідження доповідалися на звітних наукових конференціях викладачів, аспірантів і студентів Рівненського державного гуманітарного університету (2023-2024 рр.), обговорювалися на засіданнях кафедри інформаційних технологій та моделювання Рівненського державного гуманітарного університету.

Публікації. Основні результати дослідження були представлені на III Всеукраїнській науково-практичній конференції «Підготовка педагогів до професійної діяльності в умовах змішаного навчання» (м. Рівне, 28-29 травня 2024р.). У результаті опубліковано тези конференції та отримано сертифікат.

Обсяг роботи: Робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел. Перший розділ присвячений дослідженню проблеми ідентифікації образів. У другому розділі описується використання бібліотеки EMGUCV, та проблеми при обробці зображень. Третій розділ – це реалізація застосунку, використані технології та алгоритми. Четвертий розділ – інструкція користувача. Список літератури містить 20 джерел.

РОЗДІЛ 1

РОЗПІЗНАВАННЯ ОБРАЗІВ

1.1. Предмет розпізнавання образів

Розпізнавання образів є міждисциплінарною сферою, яка об'єднує наукові галузі, зосереджені на вирішенні спільної задачі – ідентифікації об'єктів, що належать певному класу, серед множини нечітких елементів, розподілених між різними класами.

Клас образів – це категорія, яка визначається спільними характеристиками, притаманними всім її елементам.

Образ представляє собою опис об'єкта як представника відповідного класу.

У ситуаціях, коли множина образів поділена на непересічні класи, для їх автоматичного розподілу доцільно застосовувати спеціалізовані пристрої.

Між класами образів і самими образами спостерігається певна ієрархічна залежність.

Наприклад, літери алфавіту та цифри можна розглядати як образи, якщо буквено-цифрові символи трактуються як класи образів. Друковані чи рукописні форми, як-от літера "А", виступають образами букви англійського алфавіту "А", яка, у свою чергу, належить до конкретного класу образів [1].

1.2. Історія формування теорії розпізнавання образів

Теорія розпізнавання образів як науковий напрямок почала формуватися наприкінці 1950-х років. Спочатку це була радше концептуальна постановка задачі, ніж її формальне визначення. Основна ідея полягала у створенні пристрою, здатного класифікувати різноманітні ситуації аналогічно до того, як це роблять живі організми. Така загальна постановка завдання стимулювала розвиток різних напрямів досліджень.

Деякі вчені зосередилися на створенні моделей процесу сприйняття. Інші вважали пріоритетним розробку алгоритмів навчання для розпізнавання образів із метою вирішення конкретних задач. Ще одна група дослідників займалася пошуком нових математичних підходів до проблеми.

На початкових етапах усі ці напрями демонстрували значний прогрес, проте з часом темпи розвитку сповільнилися. Це призвело до формування двох протилежних точок зору на проблему.

Перша точка зору передбачала, що необхідно знайти такий опис об'єктів, який базується на апріорній інформації про них, щоб процес класифікації став значно простішим. Друга позиція наголошувала на важливості пошуку ефективного правила класифікації серед множини можливих варіантів. [2].

1.3. Проблема розпізнавання образів

Загалом задача розпізнавання образів складається з двох основних етапів: навчання та розпізнавання.

Етап навчання передбачає демонстрацію окремих об'єктів із вказівкою на їхню належність до певного класу образів. У результаті система повинна навчитися однаково реагувати на всі об'єкти, які належать до одного класу, і розрізняти об'єкти, що належать до різних класів. Навчання стає основою для формування алгоритмів, які в майбутньому дозволяють системі самостійно обробляти нові дані.

Ключовим аспектом є те, що навчання має завершуватися на основі скінченної кількості прикладів, без додаткових підказок чи втручання. Це принципово важливо для розробки універсальних систем, здатних працювати в умовах, де нові об'єкти можуть відрізнятися за характеристиками від тих, які були представлені під час навчання. Об'єктами навчання можуть виступати як візуальні дані (зображення, літери), так і явища зовнішнього світу (наприклад, звуки, фізіологічні стани організму чи навіть патерни поведінки).

Після етапу навчання система переходить до розпізнавання нових об'єктів. Цей процес демонструє, наскільки успішно система засвоїла закономірності, необхідні для класифікації. Ефективність системи оцінюється за її здатністю узагальнювати набуті знання на нові, невідомі дані.

Автоматизація цих процедур є центральним завданням у проблемі навчання систем розпізнавання образів. Вона вимагає розробки ефективних алгоритмів, здатних адаптуватися до нових умов і працювати з великою кількістю різномірних даних.

Важливо зазначити, що у випадках, коли людина самостійно розробляє або адаптує правила класифікації й передає їх машині, проблема розпізнавання вирішується лише частково. Такий підхід обмежує автономність системи, адже основну й найскладнішу частину – навчання – бере на себе людина. Справді автономні системи розпізнавання образів повинні мати здатність самостійно формувати правила класифікації, спираючись на отримані дані.

Крім того, з розвитком сучасних технологій у галузі машинного навчання та штучного інтелекту дедалі більшого значення набувають методи, засновані на нейронних мережах, які дозволяють автоматизувати процес навчання навіть для дуже складних і неоднорідних даних. Такі підходи відкривають нові перспективи для створення універсальних і адаптивних систем розпізнавання образів.

1.4. Класи та їх властивості

Одним із найвдаліших формулювань ключової парадигми теорії розпізнавання є наступне: будь-який об'єкт у природі є унікальним, але всі об'єкти є типізованими. Суть цієї парадигми полягає у тому, що кожен об'єкт характеризується певними властивостями. Наявність чи відсутність цих властивостей, а також їхні якісні та кількісні характеристики розглядаються як ознаки об'єкта.

Унікальність об'єкта означає, що в природі не існує двох різних об'єктів, для яких збігаються абсолютно всі ознаки. Це дозволяє, принаймні теоретично,

відрізнити будь-який об'єкт від іншого. Разом із тим, окремі ознаки різних об'єктів можуть збігатися, що дає підстави говорити про їх належність до одного типу або класу.

Фундаментальні поняття "клас" та "об'єкт" у теорії розпізнавання неможливо повністю формалізувати. Проте можна надати їх неформальні визначення. Об'єктом називають будь-яку сутність, що існує або могла б існувати в реальному світі, а також будь-яке явище чи процес. Це визначення є дуже широким і може уточнюватися залежно від задачі. Так, Ейфелева вежа буде об'єктом у будь-якому контексті, а, наприклад, "розвиток етнокультурного стану Ефіопії та Антарктиди в період із липня 1898 року до січня 1927 року" — навряд чи.

Класом називають сукупність об'єктів, які мають спільні ознаки. Клас може об'єднувати фізично існуючі сутності (наприклад, яблуко, автомобіль) або бути абстрактним поняттям (горе, економічний крах тощо). Ознаки, які дозволяють відрізнити представників одного класу від іншого, називають інформативними, а ті, що є спільними для всіх представників класу, — інваріантами. Формальніше визначення класу звучить так: класом називають сукупність об'єктів, пов'язаних деяким відношенням еквівалентності або, у крайньому разі, толерантності. Відношення еквівалентності є симетричним, рефлексивним і транзитивним (наприклад, "дорівнювати"), а відношення толерантності, як-от "бути схожим", може не виконуватися транзитивно.

Усі представники класу мають спільний набір ознак, що впливає із визначення. Реалізації класів змінюються, адже різні об'єкти одного класу можуть відрізнятися за несуттєвими ознаками, маючи спільні інваріантні ознаки. Крім того, об'єкти можуть змінюватися з часом і переходити до іншого класу, як, наприклад, пуголовок стає жабою. Ознайомлення зі скінченною кількістю представників класу дозволяє впізнавати інших його представників. Це твердження лежить в основі навчання на прикладах.

Задача розпізнавання образів полягає у створенні систем, які здатні класифікувати об'єкти, явища чи процеси на основі їхніх ознак. Практичні

реалізації методів розпізнавання називають системами розпізнавання (СР). Такі системи дозволяють звільнити людину від рутинної роботи, забезпечують стабільність результатів і можуть діяти з високою швидкістю, що є важливим у багатьох критичних ситуаціях, наприклад, при керуванні літаком у складних умовах. Автоматизація розпізнавання дозволяє звільнити людину від одноманітних завдань, підвищити якість виконання задач та збільшити швидкість обробки інформації.

Етапи створення системи розпізнавання передбачають розподіл об'єктів на класи відповідно до вибраного принципу, розробку словника ознак, опис кожного класу мовою ознак, вибір або створення засобів визначення ознак, а також реалізацію алгоритмів, які зіставляють апостеріорні та апріорні дані для прийняття рішення.

Завдання створення систем розпізнавання мають спільний методичний підхід, незалежно від специфіки предметної області. Разом із тим, кожна система розпізнавання створюється під конкретну задачу й не є універсальною. Процес створення системи є поступовим, із використанням ітераційного уточнення її структури відповідно до нових даних. Таким чином, системи розпізнавання є важливим інструментом для вирішення задач у різних сферах, від медицини до аерокосмічної галузі, та базуються на спільних методичних засадах.

Людська система розпізнавання образів демонструє виняткову універсальність та гнучкість, але вона значно відрізняється від технічних систем розпізнавання (СР). У природних умовах людський мозок використовує складні алгоритми для класифікації та обробки інформації. Ці алгоритми дозволяють враховувати численні варіації у зовнішньому вигляді, поведінці чи контексті об'єктів і навіть створювати нові категорії на основі обмежених даних. Технічні СР, хоча й ефективні, мають суттєві обмеження. Вони зазвичай спеціалізовані на певному класі задач, наприклад, розпізнаванні захворювань серця або ідентифікації літаків, і не можуть виконувати інші задачі без додаткового налаштування.

Головною метою розробки СР є автоматизація процесів, де швидкість і точність відіграють вирішальну роль. У таких випадках, як розпізнавання ворожих об'єктів у військових системах або діагностика захворювань у медицині, технічні СР значно перевершують людські можливості завдяки здатності обробляти величезні обсяги даних у реальному часі.

Попри це, створення універсальної системи розпізнавання, аналогічної людській, залишається відкритою проблемою. Це пов'язано з необхідністю врахування специфічних властивостей кожного класу об'єктів, розробки індивідуальних алгоритмів, а також формування унікального словника ознак для кожної системи. Спроби створення універсальних рішень зазвичай обмежуються збільшенням гнучкості алгоритмів та розширенням баз даних для навчання.

Розвиток систем розпізнавання має важливе значення для механізації та автоматизації у промисловості, транспорті, медицині, освіті та інших сферах. Технічні системи можуть виконувати завдання, що вимагають великих обчислювальних ресурсів, швидких реакцій або виключають втручання людини через фізичні чи психологічні обмеження.

Одним із ключових завдань для технічних СР є зниження специфічності систем шляхом впровадження адаптивних алгоритмів та використання великих обсягів апріорної інформації. Це дозволяє створювати системи, здатні до самонавчання та роботи з широким спектром задач. З іншого боку, така універсальність збільшує складність розробки та вартість систем.

Теорія розпізнавання, як фундаментальна наука, вивчає загальні закономірності та методи створення СР, незалежно від їхньої галузі застосування. Незважаючи на специфіку кожної системи, загальні принципи залишаються незмінними. Успішність створення технічної СР залежить від точності визначення її цілей, побудови словника ознак, вибору оптимальних алгоритмів і засобів обробки даних, а також від постійного аналізу отриманих результатів для подальшого вдосконалення системи.

Таким чином, розвиток технічних систем розпізнавання не лише розширює можливості людини, а й створює основу для нових рішень у різних галузях, сприяючи прогресу науки і техніки.

Технічні системи розпізнавання образів постійно вдосконалюються завдяки досягненням у галузях штучного інтелекту, нейромереж, великих даних та обчислювальної техніки. Однак їхній розвиток стикається з багатьма викликами. Одним із головних є забезпечення адаптивності системи до нових класів об'єктів і явищ без необхідності повної її перебудови. Це питання є особливо актуальним у контексті створення автономних систем, які повинні функціонувати у непередбачуваних умовах.

Ще одним викликом є забезпечення високої точності при мінімізації помилок розпізнавання, особливо у критичних сферах, таких як медицина, транспорт чи військові технології. Розробка алгоритмів, які ефективно працюють із нечіткими чи неповними даними, залишається одним із головних завдань теорії розпізнавання. Важливим є також питання етичності, безпеки та прозорості технічних СР, зокрема у ситуаціях, де їхні рішення можуть мати значні соціальні наслідки.

Із вищесказаного можна зробити наступні висновки. Теорія розпізнавання образів має фундаментальне значення для розвитку сучасних технологій, оскільки визначає основи створення систем, здатних автоматично класифікувати об'єкти та явища. Розпізнавання базується на аналізі інформативних ознак та інваріантів класів, що дозволяє ідентифікувати об'єкти навіть за умов значної варіативності. Хоча технічні системи розпізнавання є вузькоспеціалізованими, загальна методологія їхнього створення залишається універсальною. Це дозволяє застосовувати напрацьовані підходи в різних галузях науки, техніки та промисловості.

Основним завданням розробників є підвищення універсальності технічних СР та забезпечення їхньої здатності до адаптації й самонавчання. Створення гнучких і ефективних систем розпізнавання дозволяє звільнити людину від рутинної праці, підвищити швидкість і якість виконання завдань, а також

впровадити інноваційні рішення, які недоступні за участі лише людських можливостей. Отже, теорія розпізнавання образів залишається перспективною областю досліджень, що відкриває нові горизонти для розвитку науки і техніки.

1.5. Методи розпізнавання образів

- **Лінгвістичний (синтаксичний) метод**

Цей метод базується на концепції опису образів через підобрази та їх співвідношення, використовуючи принцип загальності властивостей. Основною передумовою цього підходу є те, що образи, які належать до одного класу, мають набір спільних властивостей або ознак, що відображають їхню подібність. Ці загальні властивості можуть бути частково збережені в пам'яті системи розпізнавання.

У процесі роботи системи некласифікованому образу визначають набір ознак, які іноді кодуються, а потім порівнюються з ознаками, вже записаними в пам'ять системи. Головною задачею при цьому є виділення цих спільних властивостей, що дозволяють ідентифікувати образи, належність яких до певного класу відома.

Проте вибір методу синтезу системи не вирішує всіх проблем, таких як створення конкретної програми чи реалізації. У більшості випадків доступні образи, що представляють кожен з класів, які аналізуються. У таких ситуаціях застосовують методи розпізнавання, зокрема, **«навчання з учителем»**.

У цьому підході система навчається за допомогою адаптивних схем, орієнтуючись на відому класифікацію кожного навчального зразка. Проте в певних задачах навчальні множини невідомі. У таких випадках теж використовується підхід із «навчанням з учителем», але це вимагає глибшого аналізу класів.

Для завдань навчання без учителя потрібно досліджувати класи образів більш конкретно. Прикладами лінійного представлення є блок-схеми, технічні чи архітектурні креслення. Ці структури можна описати аналітично, що спрощує

їх реалізацію на ЕОМ. Наприклад, методи слідування за контуром або іншими траєкторними точками створюють структури, які легко представляти у вигляді **ланцюгових кодів**.

Ланцюговий код – це компактний спосіб опису образів, що дозволяє стисло представити дані з великим числом областей. Він ефективний для задач, які потребують збереження форми областей. У пам'яті комп'ютера границі областей зберігаються у вигляді лінійних списків. Цей підхід значно оптимізує обробку даних і полегшує аналіз форм [4].

- **Математичний метод**

В основу математичного підходу покладені правила класифікації, які формулюються і виводяться в рамках визначеного математичного формалізму з використанням принципів загальності властивостей і кластеризації. Коли образи деякого класу представляють собою вектори, компонентами яких є дійсні числа, то цей клас можна розглядати як кластер. Побудова системи розпізнавання, яка базується на реалізації цього принципу, визначається просторовим розміщенням окремих кластерів. Якщо кластери, що відповідають різним класам, рознесені далеко один від одного, то можна користуватися простими схемами розпізнавання, наприклад, класифікацією за принципом мінімальної відстані [4].

Інший підхід відомий як метод потенціальних функцій, відрізняється тим, що об'єднуються деякі розмиті множини, які описуються так званою потенціальною функцією. При цьому здійснюється апроксимація не розв'язуючою, а дискримінантною функцією. Її значення вираховується шляхом складання значень «потенціалу», який зменшується в міру зникання від деяких центрів, що вибираються в процесі навчання. Цей підхід є еквівалентним апроксимації дискримінантної функції за допомогою функціонального ряду.

Суть полягає в тому, що на просторі вхідних векторів X задається функція, яка називається «потенціалом». Потенціал визначається наближенням двох точок і задається як функція відстані між точками. Потенціальна функція така, що вона монотонно зменшується із збільшенням відстані.

Отже, результатом побудови є потенціальне поле, яке розбиває весь простір на дві частини: де значення додатні і від'ємні. Поверхня, на якій потенціал дорівнює нулю, називається розділяючою [5].

- **Евристичний метод**

За основу евристичного підходу взяті інтуїція і досвід людини. У цьому методі використовуються принципи перерахування членів класу і визначення загальних властивостей. Зазвичай системи, створені за допомогою евристичних методів, включають набір специфічних процедур, які розробляються для вирішення конкретних задач розпізнавання. Хоча цей підхід є важливим у практичному розпізнаванні, його обмеженість полягає у відсутності чітких загальних принципів синтезу. Розробка кожної системи вимагає специфічних прийомів, які суттєво залежать від таланту та досвіду розробників.

У багатьох випадках об'єкти одного класу мають стабільні розміри і форми. За таких умов розпізнавання може здійснюватися шляхом порівняння зображень зі зразками, які називають масками, трафаретами або еталонами. Найпростіший спосіб розпізнавання образів полягає у прямому порівнянні їх з еталонами, що зберігаються в пам'яті системи. У такій моделі кожному класу відповідає окремий еталон. При надходженні нового образу невідомого класу він порівнюється з кожним із еталонів.

Класифікація базується на попередньо визначеному критерії подібності. Якщо образ найкраще відповідає еталону певного класу, то він класифікується як належний саме до цього класу. Цей підхід ефективно використовується, наприклад, у задачах розпізнавання друкованих літер чи банківських чеків.

Однак, співставлення з еталоном має свої недоліки. Основна складність полягає у виборі такого еталону, який буде найбільш підходящим для кожного класу, а також у визначенні відповідного критерію порівняння. Ці труднощі стають особливо помітними, якщо образи одного класу можуть мати значні варіації та спотворення.

Типовим прикладом таких ускладнень є розпізнавання рукописних літер. Для вирішення цієї проблеми застосовують більш вдосконалені методи, які ґрунтуються на класифікації за набором обраних характеристик. Ці характеристики, або ознаки, є відносно нечутливими до незначних змін та спотворень.

У цьому підході розпізнавання образів поділяється на дві основні задачі. Перша – визначення, які саме ознаки повинні бути виміряні у вхідному образі. Зазвичай вибір характеристик залежить від конкретної задачі, і часто цей процес є суб'єктивним. На сьогодні загальна теорія вибору ознак все ще перебуває на стадії розвитку.

Друга задача полягає у класифікації, тобто у визначенні, до якого класу належить вхідний образ, виходячи з виміряних ознак. Цей етап включає обробку набору ознак і прийняття рішення про належність образу до певного класу [5].

1.6. Практичне застосування

Системи розпізнавання образів мають широкий спектр застосувань, охоплюючи різноманітні галузі людської діяльності. Серед найпоширеніших прикладів таких завдань можна виділити:

- **Технічна діагностика** – застосовується для виявлення несправностей в обладнанні, контролю стану механізмів, прогнозування збоїв і забезпечення надійної роботи технічних систем.
- **Медична діагностика** – використовується для аналізу медичних зображень, наприклад, рентгенограм або МРТ, а також для виявлення патологій на ранніх стадіях.
- **Стоматологія** – розпізнавання зображень зубів, ясен і щелеп для діагностики стоматологічних захворювань, планування ортодонтичного лікування, створення 3D-моделей для протезування та імплантації.

- **Розпізнавання букв** – знаходить застосування в оптичному розпізнаванні тексту (OCR), автоматичній обробці документів, а також у створенні цифрових архівів.
- **Розпізнавання штрих-кодів** – широко застосовується в логістиці, торгівлі, контролі товарообігу та автоматизації складів.
- **Розпізнавання автомобільних номерів** – необхідне для автоматизації роботи систем дорожнього руху, платних доріг, паркувань і правоохоронних органів.
- **Розпізнавання осіб та інших біометричних даних** – використовується в системах безпеки, контролю доступу, ідентифікації особистості та реалізації безконтактної аутентифікації.
- **Розпізнавання мови** – знаходить застосування в системах голосового управління, розумних помічниках, перекладах та інших інтерактивних технологіях.

Таким чином, системи розпізнавання образів є невід'ємною частиною сучасного світу, їх застосування сприяє підвищенню ефективності, точності й швидкості виконання завдань у різних сферах.

1.7. Порівняння з аналогами

У процесі пошуку аналогів програм для ідентифікації об'єктів не було знайдено жодних рішень у вільному доступі, які б відповідали вимогам конкретного дослідження. Це обумовлено тим, що подібні системи найчастіше створюються для вузькоспеціалізованих завдань і не завжди стають доступними у відкритому доступі.

Водночас існує велика кількість наукових робіт і теоретичних підходів, які описують можливі методи реалізації алгоритмів ідентифікації. Ці підходи базуються на різних методах, таких як:

- математичні моделі для аналізу даних,
- нейронні мережі для розпізнавання образів,
- методи кластеризації та статистичні підходи до обробки інформації.

Проте більшість із цих теорій є лише основою для створення програмного забезпечення і не включають готові рішення. Це підтверджує унікальність розробки власної системи ідентифікації об'єктів, яка враховує специфічні вимоги та особливості поставленого завдання [5].

Таким чином, основною перевагою даного проєкту є його адаптація до конкретних умов використання, що робить його більш цілеспрямованим і ефективним у порівнянні з теоретичними підходами або недоступними аналогами.

Висновки до першого розділу

Теорія розпізнавання образів як науковий напрямок є основою для розвитку сучасних технологій, що дозволяють автоматично класифікувати об'єкти та явища. Аналіз інформативних ознак і інваріантів класів є критичним для забезпечення точності ідентифікації об'єктів навіть за умови значної варіативності. Незважаючи на те, що технічні системи розпізнавання мають вузьку спеціалізацію, загальні методологічні підходи до їхнього створення залишаються універсальними, що дозволяє адаптувати їх до різних галузей науки, техніки та промисловості. Основним завданням для розробників є підвищення універсальності технічних систем розпізнавання та забезпечення їхньої здатності до адаптації й самонавчання. Це необхідно для підвищення ефективності систем у виконанні рутинних завдань, підвищення якості виконання завдань і впровадження інноваційних рішень. Розвиток систем розпізнавання дозволяє значно скоротити необхідність людського втручання, що є особливо важливим у критичних ситуаціях, таких як керування літаком або діагностика захворювань. Виклики залишаються щодо забезпечення адаптивності системи до нових класів об'єктів і явищ без перебудови, а також досягнення високої точності при мінімізації помилок розпізнавання, особливо у критичних сферах.

Для досягнення цих цілей необхідно зосередитися на вдосконаленні алгоритмів машинного навчання, створенні більш інформативних моделей для розпізнавання, а також обробці і зберіганні великих обсягів даних. Інтеграція мультиагентних систем, здатних обробляти й аналізувати інформацію в реальному часі, є перспективним напрямком для забезпечення високої точності й швидкості розпізнавання образів. Важливо також враховувати етичні аспекти, пов'язані з розпізнаванням образів, гарантування конфіденційності та захист персональних даних. Розробка стандартизованих підходів до тестування й оцінки точності систем розпізнавання є важливим кроком у забезпеченні їхньої надійності та ефективності.

Зростаюча роль розпізнавання образів в різних сферах життєдіяльності свідчить про його великий потенціал для вдосконалення багатьох процесів і підвищення якості життя. Впровадження технологій розпізнавання образів в стоматології, медичній діагностиці, промисловості та інших галузях відкриває нові можливості для поліпшення якості послуг і оперативності роботи. У майбутньому основним завданням буде забезпечення сталого розвитку й інтеграції цих технологій з існуючими системами, що сприятиме ефективному використанню ресурсів і підвищенню загальної ефективності.

Важливо розвивати співпрацю між академічними колами, індустрією та державними організаціями для обміну досвідом і знаннями. Таке партнерство сприятиме швидкому впровадженню новітніх технологій, розробці кращих практик і створенню інфраструктури для підтримки інновацій в області розпізнавання образів. Створення відкритих баз даних та стандартів обміну інформацією також буде сприяти підвищенню якості досліджень і розвитку спільних проектів. Наступним важливим аспектом є постійне навчання та підвищення кваліфікації фахівців, залучених до розробки й використання технологій розпізнавання образів, що забезпечить ефективну адаптацію до змінюваних умов і вимог. Технології розпізнавання образів мають великий потенціал для трансформації різних галузей, підвищення їхньої ефективності й

якості послуг. Однак для реалізації цього потенціалу необхідно забезпечити комплексний підхід, який враховує технічні, етичні та соціальні аспекти.

РОЗДІЛ 2

ОБРОБКА ЗОБРАЖЕНЬ. БІБЛІОТЕКА EMGU.CV

2.1. Обробка зображень

Обробка зображень – це комплекс методів та алгоритмів, що використовуються для аналізу, трансформації та вдосконалення зображень. Вхідними даними для такого процесу зазвичай є цифрові зображення, отримані шляхом фотографування, сканування чи запису відео. Мета обробки може бути різною: від створення естетично привабливих зображень до вилучення інформації для подальшого аналізу.

Однією з ключових задач обробки є підготовка зображень для подальшого використання. Наприклад, це може бути корекція кольору, видалення шумів, підвищення чіткості чи адаптація формату для друку або трансляції. Ці методи широко застосовуються в поліграфії, телебаченні, кінематографії та дизайні.

Інша важлива сфера застосування – це автоматичний аналіз зображень для отримання інформації. Прикладами є розпізнавання об'єктів, тексту, обчислення геометричних параметрів або моніторинг динамічних змін у відеозаписах. Такі методи є основою для систем комп'ютерного зору, біометричної ідентифікації, медичної діагностики та автоматизації наукових досліджень.

Крім того, обробка зображень охоплює роботу з динамічними даними, тобто відео, де кожен кадр є послідовністю зображень, пов'язаних у часі. У цьому випадку застосовуються додаткові методи, такі як виявлення руху, відстеження об'єктів чи стабілізація відеопотоку. Ці технології активно використовуються у сфері відеоспостереження, кінематографії та ігровій індустрії.

Сучасні інструменти та технології обробки зображень базуються на досягненнях штучного інтелекту та машинного навчання. Завдяки цьому можливості автоматизації та точність аналізу значно зросли. Наприклад,

нейронні мережі дозволяють реалізовувати такі складні завдання, як сегментація зображень, відновлення пошкоджених ділянок та покращення якості зображень з низькою роздільною здатністю.

Таким чином, обробка зображень є невід'ємною складовою багатьох сучасних галузей, сприяючи автоматизації процесів, підвищенню ефективності та точності роботи з візуальними даними [6].

2.2. Історія

Ще в середині XX століття обробка зображень здебільшого здійснювалася аналоговими методами, що базувалися на використанні оптичних пристроїв, таких як лінзи, дзеркала та фільтри. Ці методи дозволяли виконувати задачі корекції зображень, їх масштабування, повороту та інші прості маніпуляції. Зокрема, оптичні методи залишаються важливими в спеціалізованих галузях, таких як голографія, де вони забезпечують високу точність і реалістичність відображення тривимірних об'єктів.

Однак із середини XX століття стрімкий розвиток комп'ютерних технологій призвів до появи цифрових методів обробки зображень, які поступово витіснили аналогові підходи. Зростання обчислювальної потужності комп'ютерів і поява нових алгоритмів дозволили досягти значно більшої точності, гнучкості та надійності у виконанні завдань, пов'язаних із роботою з візуальною інформацією.

Цифрова обробка зображень відкрила доступ до виконання складних завдань, які раніше були неможливими або вимагали значних зусиль. Наприклад, стало можливим виконувати автоматичне видалення шумів, покращення контрастності, сегментацію зображень і навіть розпізнавання об'єктів. Окрім цього, цифрові методи відзначаються більшою простотою реалізації, оскільки можуть бути повністю автоматизовані та інтегровані у різні програмні та апаратні системи.

Сучасні цифрові системи обробки зображень використовують спеціалізоване апаратне забезпечення, таке як процесори з конвеєрною обробкою інструкцій, графічні процесори (GPU) та багатоядерні обчислювальні системи. Особливе значення такі технології мають для обробки відео, оскільки відеопотік складається з великої кількості кадрів, які потребують обробки в реальному часі. В таких системах застосовуються алгоритми стискання відео, виявлення руху та відстеження об'єктів.

Додатково, цифрова обробка зображень широко базується на програмних інструментах, які реалізують математичні методи аналізу даних. Популярними засобами є MATLAB, Mathcad, Maple та Mathematica. Ці програми надають базові функції для роботи із зображеннями, а також спеціалізовані пакети розширення, такі як Image Processing Toolbox у MATLAB. Такі інструменти дозволяють виконувати складні операції, включаючи фільтрацію, трансформації, сегментацію та аналіз зображень, із мінімальними витратами часу і зусиль.

Застосування цифрових методів обробки зображень наразі поширюється практично у всі сфери людської діяльності. Вони знаходять своє використання в медицині (для аналізу рентгенівських знімків чи МРТ), космічних дослідженнях (для обробки зображень із супутників), системах безпеки (розпізнавання облич та номерних знаків) і багатьох інших галузях. Цифрова обробка зображень не тільки замінила аналогові методи, але й суттєво розширила межі можливого у роботі з візуальними даними [6].

2.3. Види комп'ютерної графіки

У комп'ютерній графіці виділяють три основні види: растрову, векторну та фрактальну. Кожен із цих видів має свої принципи формування зображень, які визначають їхню придатність для різних задач і сфер застосування.

Растрова графіка

Растрова графіка базується на поданні зображення у вигляді сітки, кожен елемент якої називається пікселем (Picture Element). Кожен піксель має певний

колір і яскравість, що дозволяє відтворювати складні візуальні образи. Якість растрового зображення залежить від кількості пікселів (роздільної здатності), яка визначає деталізацію, та кольорової глибини, що впливає на кількість доступних кольорів.

Растрові зображення можуть створюватися кількома способами:

Ручне створення за допомогою графічних редакторів (наприклад, Adobe Photoshop або GIMP), що дозволяють малювати або редагувати зображення піксель за пікселем.

Сканування друкованих матеріалів, перетворюючи їх у цифровий формат.

Фотографування або відеозапис із використанням цифрових пристроїв, таких як камери та смартфони.

Основна перевага растрової графіки полягає у високій якості та реалістичності зображень, що робить її незамінною у фотографії, поліграфії та веб-дизайні. Однак недоліками є великий обсяг пам'яті, необхідний для зберігання таких файлів, і втрата якості при масштабуванні, коли пікселі стають помітними.

Растрові формати, такі як JPEG, PNG і GIF, є стандартом у мережі Інтернет завдяки їхній універсальності та сумісності з веб-технологіями.

Векторна графіка

На відміну від растрової, векторна графіка базується на математичному описі зображення. Основним її елементом є лінія, яка описується параметрами, такими як координати, товщина, колір, кривизна тощо. З цих базових елементів створюються складніші графічні об'єкти, які можна комбінувати, редагувати або трансформувати незалежно один від одного.

Однією з основних переваг векторної графіки є можливість масштабування без втрати якості. Це досягається за рахунок того, що зображення формується заново при кожному масштабуванні, використовуючи його математичний опис. Крім того, векторні файли займають менше пам'яті, ніж растрові, оскільки містять лише набір параметрів, а не дані про кожен піксель.

Векторну графіку широко використовують у створенні схем, креслень, логотипів, ілюстрацій та інших об'єктів, які потребують точності та гнучкості. Типовими форматами векторної графіки є SVG, AI (Adobe Illustrator) та EPS.

Фрактальна графіка

Фрактальна графіка базується на принципах математичного моделювання. Зображення у цьому виді графіки створюються за допомогою фракталів – геометричних об'єктів, що характеризуються самоподібністю на різних масштабах. Основою для створення фрактальних зображень є математичні формули, які задають їхню структуру.

Створення фрактальних композицій не вимагає традиційного малювання: достатньо запрограмувати формулу та параметри, після чого зображення генерується автоматично. Фрактальну графіку використовують у численних сферах, включаючи створення візуальних ефектів, художніх композицій, генерацію ландшафтів для відеоігор та моделювання природних об'єктів.

Фрактальна графіка знайшла своє місце в науці, мистецтві та розважальній індустрії завдяки унікальній здатності створювати складні та естетично привабливі зображення на основі простих математичних принципів.

2.4. Основні методи обробки зображень

Методи обробки одновимірних сигналів, такі як медіанний фільтр, часто застосовують і до двовимірних сигналів, якими є зображення. Проте з переходом до двовимірних сигналів багато з цих методів стають значно складнішими. Наприклад, у двовимірному просторі виникають нові поняття, такі як зв'язність (визначення взаємозв'язку між елементами зображення) та ротаційна інваріантність (незалежність результату обробки від обертання зображення), які є специфічними для роботи із зображеннями.

Одним із ключових інструментів у обробці сигналів є перетворення Фур'є, яке дозволяє переходити від просторової області до частотної, що дає змогу аналізувати структуру зображення за його частотними компонентами. Також

широко використовуються вейвлет-перетворення та фільтр Габора, які дозволяють локалізувати особливості зображення у просторі та частоті, що є важливим для аналізу складних структур.

Обробка зображень поділяється на два основні підходи:

Обробка у просторовій області, яка включає маніпуляції із яскравістю, корекцію гами, фільтрацію та інші перетворення, що виконуються безпосередньо над пікселями зображення.

Обробка у частотній області, яка передбачає аналіз і модифікацію спектральних характеристик зображення за допомогою перетворень, таких як перетворення Фур'є.

Дискретне перетворення Фур'є для зображень, представлених у вигляді дискретних функцій просторових координат, має важливу властивість: його результат є періодичним по просторових частотах із періодом 2π . Це забезпечує можливість виявлення частотних компонентів, характерних для структури зображення, що є особливо корисним у завданнях фільтрації, компресії або сегментації зображень [6].

Застосування сучасних методів обробки сигналів, таких як вейвлет-перетворення, дозволяє аналізувати зображення на різних масштабах, що є корисним для виявлення деталей або глобальних структур. Використання фільтра Габора дає можливість виділити орієнтовані текстурні елементи, що робить цей інструмент особливо популярним у задачах розпізнавання образів.

Таким чином, обробка зображень, використовуючи як традиційні підходи, так і сучасні трансформаційні методи, надає широкі можливості для аналізу, модифікації та автоматизованої інтерпретації візуальної інформації.

2.5. EMGUCV

EMGUCV – це бібліотека комп'ютерного зору з відкритим кодом, яка надає потужні інструменти для обробки зображень, аналізу вмісту та виконання чисельних обчислень. Вона забезпечує широкий спектр функцій для реалізації

як базових операцій, так і складних алгоритмів, що знаходять застосування в різних галузях, таких як робототехніка, автоматизація, безпека та медична діагностика.

Основні можливості EMGUCV включають:

Розпізнавання об'єктів: визначення облич, людських фігур, тексту або інших специфічних елементів на зображеннях і відео.

Аналіз руху: відстеження динаміки об'єктів у реальному часі, що використовується в системах спостереження або інтерактивних додатках.

Перетворення зображень: зміна розмірів, кольорових моделей, застосування фільтрів і корекцій для покращення якості або підготовки до подальшого аналізу.

Методи машинного навчання: інтеграція алгоритмів класифікації, кластеризації та прогнозування для вирішення завдань комп'ютерного зору.

Виявлення схожих елементів: пошук спільних рис на різних зображеннях, що корисно для аналізу збігів та ідентифікації.

Розроблена компанією Intel, бібліотека стала основою для багатьох проєктів і нині підтримується такими організаціями, як Willow Garage та Itseez, що забезпечує її актуальність і розвиток. Її сирцевий код написаний мовою C# і розповсюджується під ліцензією BSD, що дає можливість вільного використання як у навчальних, так і в комерційних цілях.

EMGUCV підтримує інтеграцію з різноманітними мовами програмування, серед яких Python, Java, Ruby, Matlab, Lua та інші. Це значно розширює спектр її використання, дозволяючи розробникам легко інтегрувати функціонал бібліотеки у свої програми незалежно від обраної платформи чи середовища.

Завдяки своїм можливостям EMGUCV активно застосовується у таких сферах:

Відеоспостереження: створення систем безпеки з функціями виявлення руху, розпізнавання осіб і небезпечних об'єктів.

Робототехніка: аналіз зображень і відео для навігації роботів у просторі.

Медицина: автоматизований аналіз зображень для діагностики, наприклад, рентгенівських знімків чи ультразвукових сканів.

Розширена реальність: виявлення і обробка елементів для інтеграції з віртуальним контентом у реальному часі.

У поєднанні з іншими сучасними технологіями EMGUCV дозволяє розробляти інноваційні рішення, що задовольняють вимоги сучасного ринку, сприяючи розвитку цифрових технологій і автоматизації [7].

2.5.1. Історія EMGUCV

Проект EMGUCV було офіційно започатковано у 1999 році з ініціативи Intel Research з метою створення інструменту, що сприяв би розвитку ресурсоемних додатків для центральних процесорів (CPU). Основними учасниками, які зробили вагомий внесок у цей проєкт, стали Intel's Performance Library Team та експерти в галузі чисельної оптимізації з підрозділу Intel.

На початкових етапах розробки бібліотека мала три ключові завдання:

1. **Сприяти дослідженням у сфері комп'ютерного зору**, забезпечуючи розробників добре оптимізованим та відкритим кодом.
2. **Популяризувати знання в галузі комп'ютерного зору**, створюючи спільну інфраструктуру для програмістів. Це дозволяло не лише вдосконалювати код, але й полегшувати його розуміння і обмін між спільнотами розробників.
3. **Розвивати комерційні додатки, засновані на технологіях комп'ютерного зору**, шляхом створення незалежної від платформи, безкоштовної та високопродуктивної бібліотеки. Ліцензійна модель дозволяла використовувати бібліотеку в комерційних додатках без обов'язкової відкритості їхнього коду.

Перша альфа-версія EMGUCV була презентована на конференції IEEE з комп'ютерного зору та розпізнавання образів у 2000 році. Протягом 2001–2005 років було випущено п'ять бета-версій, що поступово вдосконалювали

функціонал бібліотеки. У 2006 році з'явилася версія 1.0, яка ознаменувала початок стабільного використання бібліотеки в академічних і комерційних проєктах.

У 2008 році EMGUCV отримала підтримку від компанії Willow Garage, що стало новим етапом її активного розвитку. У жовтні того ж року була випущена попередня версія 1.1, яка включала в себе важливі оновлення і вдосконалення.

Другий великий випуск EMGUCV, версія 2.0, відбувся у жовтні 2009 року. Цей реліз включав значні зміни в інтерфейсі C++, спрямовані на створення простіших і безпечніших типів даних, підвищення швидкодії бібліотеки, особливо на багатоядерних системах, а також впровадження нових функцій.

З 2009 року офіційні релізи бібліотеки почали виходити з регулярністю кожні шість місяців. Подальшу розробку здійснювала незалежна команда з Росії, що отримувала підтримку від комерційних організацій.

У серпні 2012 року підтримку EMGUCV було передано некомерційній організації EMGUCV.org, яка продовжила розвиток бібліотеки з акцентом на її доступність для широкого кола користувачів та інтеграцію новітніх технологій.

Завдяки своїй відкритості та адаптованості EMGUCV стала одним із ключових інструментів у сфері комп'ютерного зору, що забезпечує ефективність розробки програмного забезпечення як у наукових дослідженнях, так і в комерційних проєктах [7].

2.5.2. Застосування

Бібліотека EMGUCV містить понад 2500 оптимізованих алгоритмів, що забезпечують потужний інструментарій для вирішення задач машинного навчання та комп'ютерного зору. Вона охоплює як класичні, так і сучасні алгоритми, що знайшли практичне застосування у різних галузях.

Алгоритми EMGUCV активно використовуються у таких сферах:

Аналіз та обробка зображень – включає базові операції обробки, такі як фільтрація, зміна яскравості, сегментація та інше.

Системи розпізнавання обличчя – ідентифікація людей на основі фотографій або відео.

Ідентифікація об'єктів – пошук і класифікація об'єктів у зображеннях та відеопотоках.

Розпізнавання жестів – аналіз руху рук або інших частин тіла у реальному часі.

Відслідковування переміщення камери – обчислення траєкторії руху камери для стабілізації відео чи створення панорам.

Побудова 3D моделей об'єктів – реконструкція об'єктів у тривимірному просторі на основі стереозображень.

Створення 3D хмар точок зі стереокамер – відтворення тривимірної сцени для інженерних чи дизайнерських цілей.

Склеювання зображень – автоматична зшивка кількох зображень для створення панорам або зображень високої роздільної здатності.

Системи взаємодії людини з комп'ютером – інтеграція комп'ютерного зору для розпізнавання рухів, жестів чи команд.

Пошук схожих зображень – швидкий аналіз великих баз даних для виявлення подібних зображень.

Усування ефекту червоних очей – автоматична корекція фотографій для покращення їх якості.

Стеження за рухом очей – вивчення траєкторій погляду для медичних чи маркетингових досліджень.

Аналіз руху – визначення траєкторій і швидкості руху об'єктів у відеопотоках.

Ідентифікація об'єктів – детальніший аналіз властивостей та класифікація об'єктів у зображенні.

Сегментація зображень – виділення окремих частин зображення, наприклад, для медичних діагностик чи аналізу карт.

Трекінг відео – відстеження траєкторії об'єктів у русі.

Розпізнавання елементів сцени – виявлення специфічних об’єктів чи зон на сцені, зокрема для додатків доповненої реальності.

Завдяки своїй функціональності EMGUCV знаходить застосування у багатьох інноваційних галузях: від медичних технологій до автоматизації виробничих процесів, а також у наукових дослідженнях та розробці розважальних додатків. Бібліотека забезпечує розробників не лише інструментами для аналізу, а й потужними засобами для впровадження сучасних технологій у повсякденне життя [8].

Висновки до другого розділу

Обробка зображень охоплює методи та алгоритми, призначені для аналізу, трансформації та вдосконалення візуальної інформації. Вона включає завдання, такі як корекція кольору, видалення шумів, підвищення чіткості та адаптація формату зображень для друку або трансляції. Ці методи застосовуються у поліграфії, телебаченні, кінематографії та дизайні. Крім того, автоматичний аналіз зображень включає розпізнавання об’єктів, тексту, обчислення геометричних параметрів і моніторинг динамічних змін у відеозаписах. Сучасні технології обробки зображень, що базуються на досягненнях штучного інтелекту та машинного навчання, значно підвищують автоматизацію процесів і точність аналізу.

Розвиток обробки зображень пройшов шлях від аналогових методів, що використовувалися у XX столітті, до цифрових підходів, які нині витіснили свої попередники завдяки зростанню обчислювальної потужності комп’ютерів і появі нових алгоритмів. Цифрові методи дозволяють виконувати складні завдання, такі як автоматичне видалення шумів, покращення контрастності, сегментація зображень і навіть розпізнавання об’єктів. Спеціалізоване апаратне забезпечення та програмні інструменти, такі як MATLAB і Image Processing Toolbox, стали основою для обробки зображень, що знайшла своє застосування в широкому спектрі галузей, включаючи медицину, космічні дослідження і системи безпеки.

У комп'ютерній графіці виділяють три основні види: растрову, векторну та фрактальну. Растрова графіка відтворює зображення за допомогою пікселів і застосовується в поліграфії та веб-дизайні. Векторна графіка дозволяє масштабувати зображення без втрати якості і часто використовується у створенні схем, логотипів і ілюстрацій. Фрактальна графіка базується на математичному моделюванні і широко застосовується у візуальних ефектах та моделюванні природних об'єктів.

Основні методи обробки зображень включають як традиційні перетворення, такі як перетворення Фур'є, так і сучасні методи, як вейвлет-перетворення та фільтр Габора. Обробка зображень поділяється на просторову і частотну області. Кожен з цих підходів дозволяє здійснювати маніпуляції з яскравістю, контрастністю, фільтрацією та іншими параметрами зображення. Використання сучасних трансформаційних методів, таких як вейвлет-перетворення, дозволяє аналізувати зображення на різних масштабах і виявляти деталі або глобальні структури.

EMGUCV є бібліотекою комп'ютерного зору з відкритим кодом, яка надає потужні інструменти для обробки зображень і аналізу вмісту. Вона підтримує розпізнавання об'єктів, аналіз руху, перетворення зображень і методи машинного навчання для вирішення завдань комп'ютерного зору. EMGUCV активно застосовується у сфері відеоспостереження, робототехніки, медицини та розширеної реальності. Історично бібліотека була розроблена Intel Research для підтримки ресурсномістких додатків, і згодом знайшла широке використання завдяки відкритому коду та підтримці таких організацій, як Willow Garage та Itseez.

Історія розвитку EMGUCV розпочалася у 1999 році як ініціатива Intel Research з метою створення інструменту для підтримки ресурсномістких додатків для центральних процесорів. Основні завдання включали сприяння дослідженням у сфері комп'ютерного зору, створення спільної інфраструктури для програмістів та розробку комерційних додатків на базі комп'ютерного зору.

Перша альфа-версія була представлена у 2000 році, а стабільна версія 1.0 вийшла у 2006 році. Згодом EMGUCV отримала підтримку від Willow Garage, що сприяло її подальшому розвитку та популяризації серед розробників.

Обробка зображень і бібліотека EMGU.CV відіграють ключову роль у сучасних технологіях комп'ютерного зору, забезпечуючи можливості для аналізу вмісту зображень, розпізнавання об'єктів і моніторингу динамічних змін. Застосування традиційних і новітніх методів обробки зображень дозволяє здійснювати складні маніпуляції з візуальною інформацією, покращуючи якість зображень і ефективність їхнього використання у різних сферах. EMGU.CV, завдяки своєму відкритому коду та підтримці спільноти розробників, надає потужні інструменти для створення інноваційних рішень у таких галузях, як робототехніка, медицина, безпека і розширена реальність. Бібліотека активно розвивається і адаптується до нових викликів, забезпечуючи можливості для вирішення складних задач комп'ютерного зору на основі сучасних методів машинного навчання та штучного інтелекту.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ЗАСТОСУНКУ

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення (Integrated Development Environment, IDE) та ряд інших інструментальних засобів, спрямованих на забезпечення всебічного процесу розробки програм. Ця платформа надає розробникам можливість створювати різноманітні типи програмного забезпечення, включаючи консольні програми, програми з графічним інтерфейсом користувача (GUI), веб-сайти, веб-застосунки, мобільні додатки та серверні компоненти.

Особливістю Visual Studio є її підтримка широкого спектра технологій і платформ. Наприклад, розробники можуть створювати програми для операційних систем Windows (Windows Desktop, Windows Server), мобільних платформ (Windows Mobile, Windows Phone) та вбудованих систем (Windows CE). Крім того, IDE дозволяє розробляти додатки для .NET Framework і .NET Compact Framework, які забезпечують гнучкість та масштабованість у середовищі керованого коду. Visual Studio також інтегрується з технологіями Microsoft Silverlight для розробки багатофункціональних веб-додатків.

Visual Studio підтримує роботу з багатьма мовами програмування, включаючи C++, C#, Visual Basic, JavaScript, Python та F#. Завдяки цьому, вона слугує універсальним інструментом як для новачків, так і для досвідчених розробників. Серед ключових функцій середовища можна виділити потужний редактор коду, інструменти налагодження, підтримку розподіленого контролю версій, інтеграцію з системами управління життєвим циклом програмного забезпечення (ALM), а також можливості для створення автоматизованих тестів.

У рамках екосистеми Visual Studio доступні такі додаткові продукти, як Visual Studio Code — легке кросплатформне середовище розробки, орієнтоване на веб-розробку та хмарні технології, а також Visual Studio for Mac, адаптоване

для розробки програм під macOS. Завдяки інтеграції з Microsoft Azure розробники отримують доступ до хмарних ресурсів, що дозволяє створювати, тестувати та розгортати масштабовані рішення безпосередньо з IDE.

Visual Studio також пропонує багатий набір розширень, доступних через Visual Studio Marketplace, що дозволяє розширити функціональність середовища відповідно до потреб користувачів. Це робить платформу Visual Studio ключовим інструментом у сучасній розробці програмного забезпечення, здатним забезпечити високу продуктивність та якість продуктів[9].

3.1. Технологія WPF

Технологія WPF (Windows Presentation Foundation) є сучасною частиною екосистеми платформи .NET, яка забезпечує інструментарій для створення багатофункціональних графічних інтерфейсів користувача (GUI) на основі сучасних стандартів і технологій[18]. Її основна перевага — використання потужностей графічного процесора через DirectX, що забезпечує високоефективний рендеринг як простих, так і складних графічних елементів.

У традиційних додатках, побудованих на основі WinForms, за відображення елементів управління та графіки відповідали такі підсистеми ОС Windows, як User32 та GDI+. Водночас WPF працює безпосередньо через DirectX, що дозволяє виконувати апаратне прискорення рендерингу графіки. Це дає змогу розробникам створювати інтерфейси, що не тільки виглядають сучасно, але й функціонують із високою продуктивністю, незалежно від складності елементів. Наприклад, апаратне прискорення використовується як для відображення кнопок, текстових полів, так і для складних 3D-об'єктів, а також анімаційних ефектів.

Однією з найвагоміших переваг WPF є підтримка мови декларативної розмітки XAML (eXtensible Application Markup Language), яка базується на стандарті XML. Використання XAML спрощує розробку інтерфейсу, дозволяючи чітко відокремити логіку бізнес-процесів (написану на C# або

VB.NET) від дизайну. Розробники можуть створювати графічні інтерфейси як повністю декларативно, так і комбінуючи код із XAML, що забезпечує високу гнучкість і зручність розробки. Крім того, XAML дозволяє розробникам створювати анімації, переходи, візуальні ефекти та адаптивні макети без значного ускладнення коду програми.

Ще однією важливою перевагою WPF є його можливості для створення сучасного інтерфейсу із застосуванням шаблонів (DataTemplates і ControlTemplates), стилів та ресурсів. Це дає змогу забезпечити єдиний візуальний стиль для всієї програми, легко підтримуючи масштабованість і повторне використання елементів. У поєднанні з широкими можливостями для роботи з мультимедіа, зокрема відтворенням аудіо та відео, WPF стає потужним інструментом для створення інтерактивних додатків.

Крім того, WPF підтримує створення програм для декількох платформ .NET, дозволяючи створювати як настільні додатки, так і більш адаптивні рішення, що відповідають сучасним вимогам користувачів. Завдяки цим особливостям Windows Presentation Foundation є невід'ємною складовою для розробки сучасного програмного забезпечення на платформі Microsoft .NET[10].

3.1.1. Переваги WPF

- Використання традиційних мов .NET-платформи

WPF надає розробникам можливість використовувати знайомі мови програмування, такі як C# та VB.NET, для створення логіки додатка. Завдяки цьому розробка стає інтуїтивно зрозумілою для тих, хто вже працював з платформою .NET, забезпечуючи швидкий старт і ефективний процес розробки.

- Декларативне визначення графічного інтерфейсу

Мова XAML, на якій базується WPF, дозволяє декларативно створювати інтерфейси користувача, спрощуючи процес проєктування. Вона дає змогу розробляти складні графічні інтерфейси без написання значної кількості коду.

Крім того, поєднання XAML із C# або VB.NET забезпечує гнучкість і можливість легко оновлювати дизайн і функціональність додатка [10].

- Адаптивність до роздільної здатності екрану

Одиниці виміру в WPF є незалежними від пристрою, що забезпечує масштабованість інтерфейсу для різних розмірів екранів і роздільних здатностей. Це робить додатки більш адаптивними, зручними для використання на пристроях із різною щільністю пікселів.

- Розширені функціональні можливості

WPF відкриває доступ до нових функцій, які раніше були важкодоступними або зовсім недоступними в WinForms. Наприклад, розробники можуть легко створювати тривимірні моделі, використовувати прив'язку даних (data binding), стилі, шаблони, теми та інші сучасні інструменти для проєктування додатків.

- Сумісність із WinForms

WPF дозволяє інтегрувати традиційні елементи управління WinForms, що забезпечує зворотну сумісність і можливість використання раніше створених компонентів у нових WPF-додатках.

- Багатофункціональність

WPF надає широкий набір інструментів для створення мультимедійних програм, двовимірної та тривимірної графіки, анімацій, а також багатий набір вбудованих елементів управління. Крім того, розробники можуть створювати власні елементи управління, що забезпечує додаткову гнучкість.

- Апаратне прискорення графіки

Усі графічні елементи в WPF рендеряться через Direct3D, що дозволяє використовувати можливості графічного процесора відеокарти. Це не тільки підвищує продуктивність, але й робить відтворення графіки, тексту чи анімацій більш плавним.

- Кросплатформенність у межах Windows

Додатки, створені на WPF, сумісні з широким спектром операційних систем сімейства Windows, починаючи від Windows XP і до найновіших версій,

таких як Windows 10 і Windows 11. Це забезпечує високу гнучкість і доступність розроблених програм.

Обмеження WPF

Незважаючи на значні переваги, WPF має і деякі недоліки:

- **Обмеження у створенні ігор**

Для додатків із великою кількістю тривимірних об'єктів, таких як ігри, WPF менш ефективний порівняно з DirectX або спеціалізованими фреймворками, наприклад, Monogame чи Unity.

- **Ресурсомісткість**

Додатки, створені на WPF, зазвичай займають більше пам'яті та мають більший обсяг файлів порівняно з аналогами на WinForms. Однак це компенсується розширеними графічними можливостями та більш ефективною роботою з графікою [11].

3.1.2. Архітектура WPF

Архітектура Windows Presentation Foundation (WPF) побудована на дворівневій моделі: **керованому API (Managed API)** і **некерованому API (Unmanaged API)**, який забезпечує інтеграцію з DirectX.

Керований API

Керований API виконується під управлінням середовища виконання .NET (Common Language Runtime, CLR) і є основою платформи WPF. Цей рівень містить наступні ключові компоненти:

- **PresentationFramework.dll**: реалізує основні компоненти та елементи управління, необхідні для побудови графічного інтерфейсу користувача.
- **PresentationCore.dll**: містить базові типи для компонентів PresentationFramework.dll, забезпечуючи низькорівневу функціональність для рендерингу графіки.

- **WindowsBase.dll**: надає допоміжні класи, які можуть використовуватися як у WPF, так і поза його межами.

Некерований API

Некерований API забезпечує низькорівневу інтеграцію з DirectX, використовуючи наступні компоненти:

- **milcore.dll**: відповідає за зв'язок між компонентами WPF і DirectX, реалізуючи рендеринг графіки. Цей модуль написаний на C++ для ефективної роботи з апаратним забезпеченням.
- **WindowsCodecs.dll**: бібліотека, що забезпечує підтримку роботи із графічними форматами (JPEG, PNG, BMP та інші), необхідними для обробки зображень у WPF.

Інтеграція з компонентами операційної системи

На нижньому рівні архітектури розташовані компоненти операційної системи Windows і DirectX:

- **Direct3D**: відповідає за трансляцію графічних об'єктів WPF у команди, зрозумілі для відеокарти, забезпечуючи апаратне прискорення графіки.
- **user32.dll**: використовується для виконання задач, не пов'язаних із візуалізацією, зокрема для обробки подій взаємодії з користувачем.

Розвиток WPF

WPF є частиною платформи .NET, і його версії розвиваються паралельно з фреймворком. Перша версія WPF 3.0 вийшла разом із .NET Framework 3.0 у 2006 році, супроводжуючи запуск операційної системи Windows Vista.

Подальші оновлення включали поліпшення продуктивності, розширення функціональних можливостей і інтеграцію з сучасними технологіями. Наприклад, WPF 4.6, випущений у липні 2015 року, представив нові можливості для роботи з графікою та мультимедіа, а також удосконалення в управлінні даними.

Особливості дворівневої архітектури

Дворівнева модель архітектури WPF забезпечує:

- **Продуктивність:** використання Direct3D дозволяє оптимізувати рендеринг і забезпечує плавність графіки.
- **Гнучкість:** поєднання високорівневого API .NET і низькорівневої інтеграції з DirectX відкриває широкі можливості для створення додатків.
- **Масштабованість:** завдяки модульній структурі платформи додатки можуть легко адаптуватися до змінних потреб.

WPF залишається одним із найсучасніших і найефективніших інструментів для розробки настільних додатків із багатим графічним інтерфейсом і розширеними мультимедійними можливостями [12].

3.2. Технологія JSON

Для зберігання оброблених зображень у розробленому додатку використовується технологія JSON (JavaScript Object Notation) — текстовий формат обміну даними, створений для взаємодії із JavaScript-кодом. JSON став стандартним форматом обміну даними, що активно використовується у веб-застосунках, зокрема у службах ASP.NET AJAX, які створюються в середовищі Windows Communication Foundation (WCF) [19].

JSON може використовуватися не лише для AJAX-служб, інтегрованих із ASP.NET, але й для створення окремих служб, що підтримують цей формат даних. У таких випадках JSON можна використовувати як альтернативу XML.

Для перетворення об'єктів .NET у JSON-дані і навпаки використовується `DataContractJsonSerializer`. Цей інструмент підтримує більшість типів .NET, включаючи примітивні типи, масиви, колекції та складні типи з атрибутами `DataContractAttribute` і `DataMemberAttribute`.

JSON підтримує поліморфну серіалізацію, тобто похідні типи можуть бути серіалізовані там, де очікується базовий тип.

Під час десеріалізації в базовий тип інформація про конкретний тип може бути втрачена. Наприклад, якщо рядок JSON серіалізувався як тип `Uri`, то після десеріалізації в базовий тип буде повернуто `String`.

Тип даних JSON (наприклад, число чи рядок) може не збігатися з відповідним .NET-типом, але десеріалізація все одно буде успішною, якщо формат даних дозволяє це зробити.

Для серіалізації користувацьких типів, таких як похідні класи, можна використовувати атрибути `KnownTypeAttribute` або `ServiceKnownTypeAttribute`. Наприклад, якщо є базовий клас `Animal` і похідний клас `Cat`, необхідно вказати у відповідному атрибуті, що `Cat` є похідним типом [19].

JSON ефективно використовується для зберігання інформації про оброблені зображення завдяки таким перевагам:

Легкий формат: компактний текстовий вигляд дозволяє легко читати і обробляти дані.

Сумісність із .NET: серіалізація та десеріалізація здійснюється без втрати даних і з мінімальними витратами ресурсів.

Простота інтеграції: JSON легко інтегрується з іншими технологіями та платформами, забезпечуючи обмін даними між різними компонентами програми.

На рисунку 3.1 представлено приклад класу для серіалізації та десеріалізації даних, що стосуються зображень. Клас включає методи для перетворення об'єктів у JSON-формат і зворотного відновлення об'єктів із збережених даних.

Таким чином, використання JSON у розробленому додатку забезпечує швидку та надійну роботу з даними, спрощуючи управління обробленими фотографіями та їх інтеграцію з іншими частинами системи [13].

```

using System.Drawing;

namespace TrassogicalTool
{
    4 references
    public class ProcessedImage
    {
        2 references
        public Point Center { get; set; }

        2 references
        public string Path { get; set; }
    }
}

```

Рис. 3.1. Клас для JSON

3.3. Інсталяція MS Visual Studio і його налаштування

Для інсталяції Visual Studio потрібно завантажити інсталяційний файл з офіційного сайту Unity (<https://visualstudio.microsoft.com/vs/>). Перед завантаженням розробнику потрібно вирішити: завантажувати безкоштовну, комерційну чи професійну версію.

Після того, як розробник вирішив, він має обрати потрібне поле на сайті. Найкращим варіантом для студентів і початківців, а також для всіх, хто вже має досвід роботи з середовищами розробки, але тільки починає працювати з Visual Studio, пропонується завантажити безкоштовну версію.

При першому завантаженні розробнику надається місяць безкоштовного використання усіх функцій, за який розробнику надається змога зрозуміти, потрібні йому платні функції, чи ні. Отже, в цій роботі буде використовуватися безкоштовна версія. І для її завантаження потрібно натиснути на поле, виділене на (див. рис. 3.2.), яке знаходиться на сайті, вказаному нижче. Поки файл завантажується потрібно зареєструвати аккаунт на сайті.

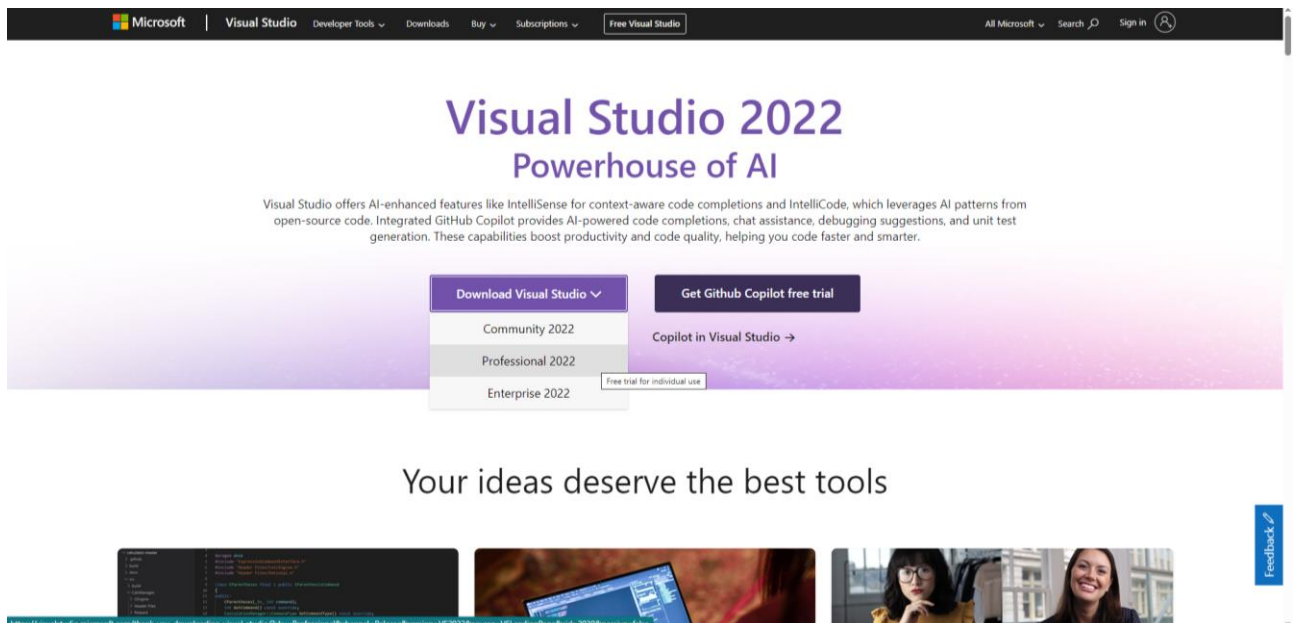


Рис. 3.2. Вибір потрібної версії

Після завантаження, розробник запускає файл і проходить усі кроки інсталяції, після завершення якої, буде відкрито вікно. У нього потрібно ввести дані з акаунту, який зареєстрували на сайті (див. рис. 3.3.). Підсумувавши все, написане вище, інсталяцію можна розбити на такі кроки:

1. Завантаження інсталяційного файла.
2. Реєстрація на сайті Microsoft.
3. Інсталяція завантаженого файла.
4. Активація програми.

Після виконання усіх цих кроків, інсталяцію програми можна вважати завершеною.

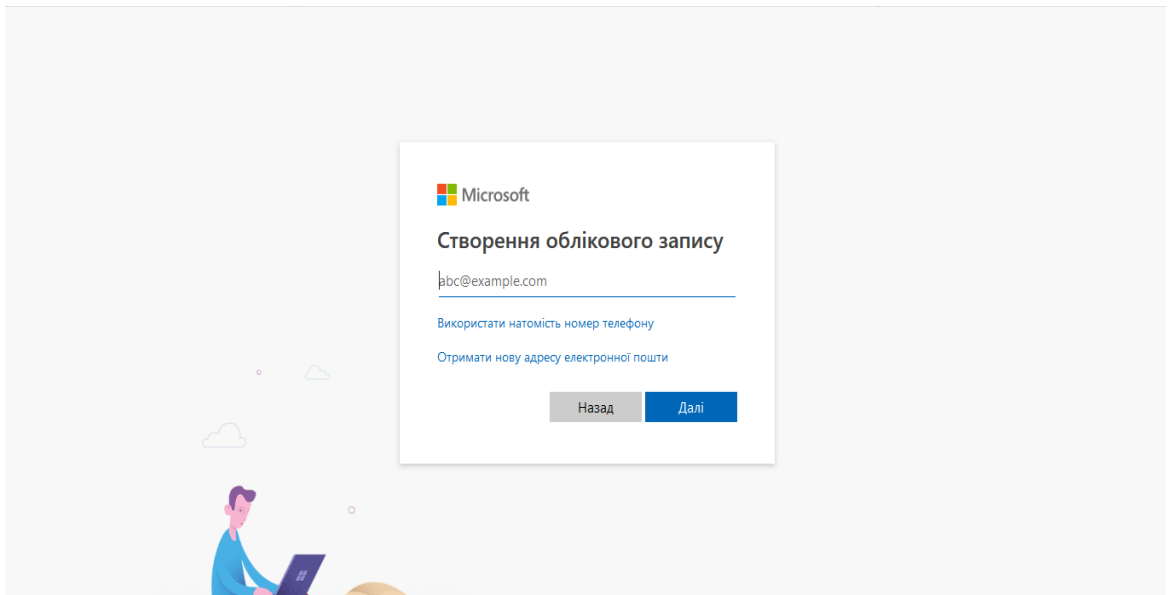


Рис. 3.3. Сайт реєстрації акаунта

3.4. Початок роботи та ознайомлення з інтерфейсом

Після завершення інсталяції можна приступати до роботи з ігровим рушієм. Одразу після запуску з'являється вікно, у якому можна побачити проекти, з якими працювали, а також кнопки для відкриття або створення нового проекту (див. рис. 3.4).

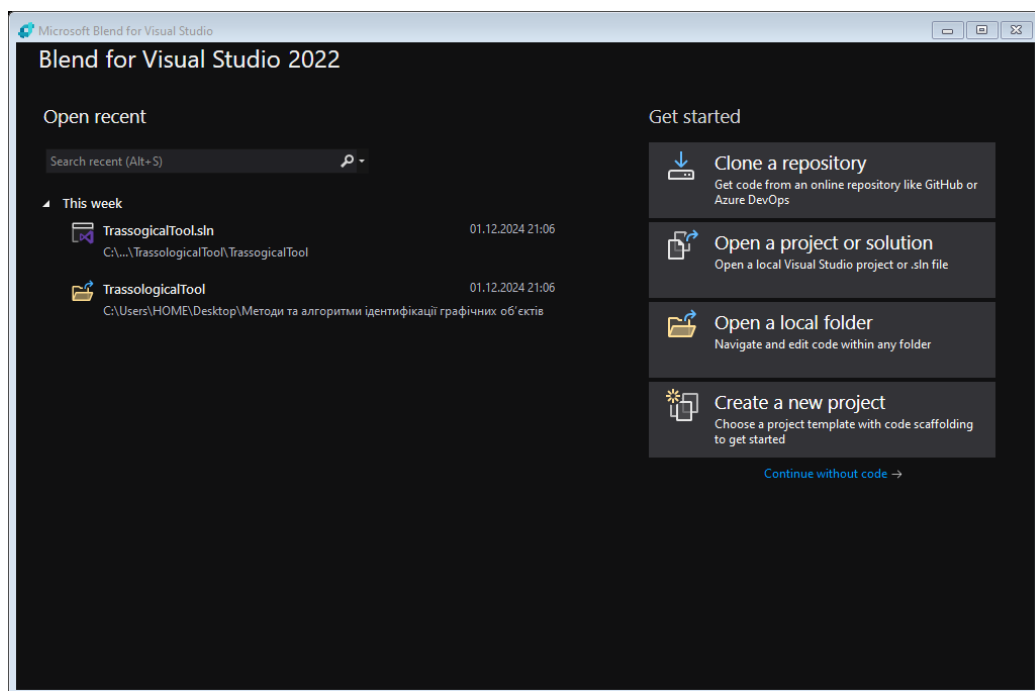


Рис. 3.4. Початкове вікно Visual Studio

Якщо користувач обирає проект, з яким він працював, він одразу відкривається. Якщо було обрано новий проект, то з'являється нове вікно, у якому треба дати проекту назву, обрати місце його збереження і обрати тип застосування. Після цього кроку з'являється вікно з головним робочим інтерфейсом (див. рисунок 3.5).

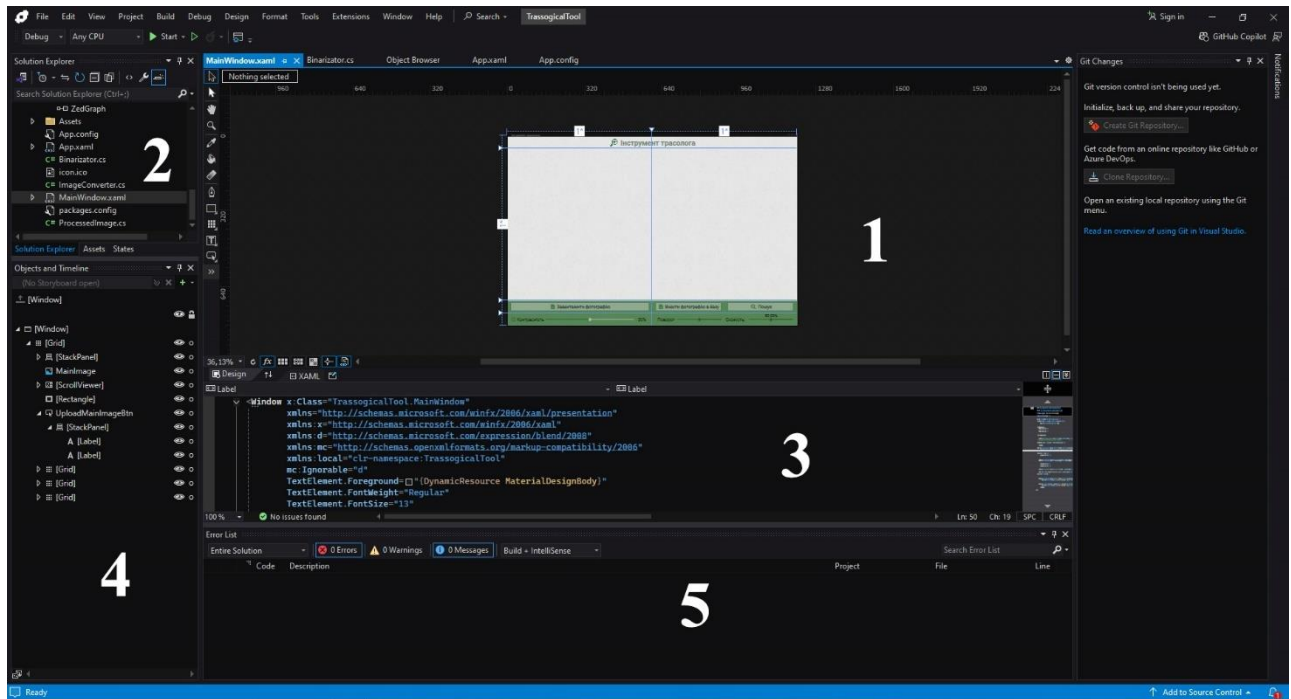


Рис. 3.5. Головний інтерфейс Visual Studio

Головний інтерфейс поділено на 5 зон:

- 1-зона. Відображає зовнішній вигляд вікна додатку.
- 2-зона. Файлова структура додатку.
- 3-зона. Код XAML для відображення всього, що зображено у зоні № 1.
- 4-зона. Властивості застосування.
- 5-зона. Вивід помилок та попереджень.

3.4.1. Під'єднання бібліотеки EMGUCV

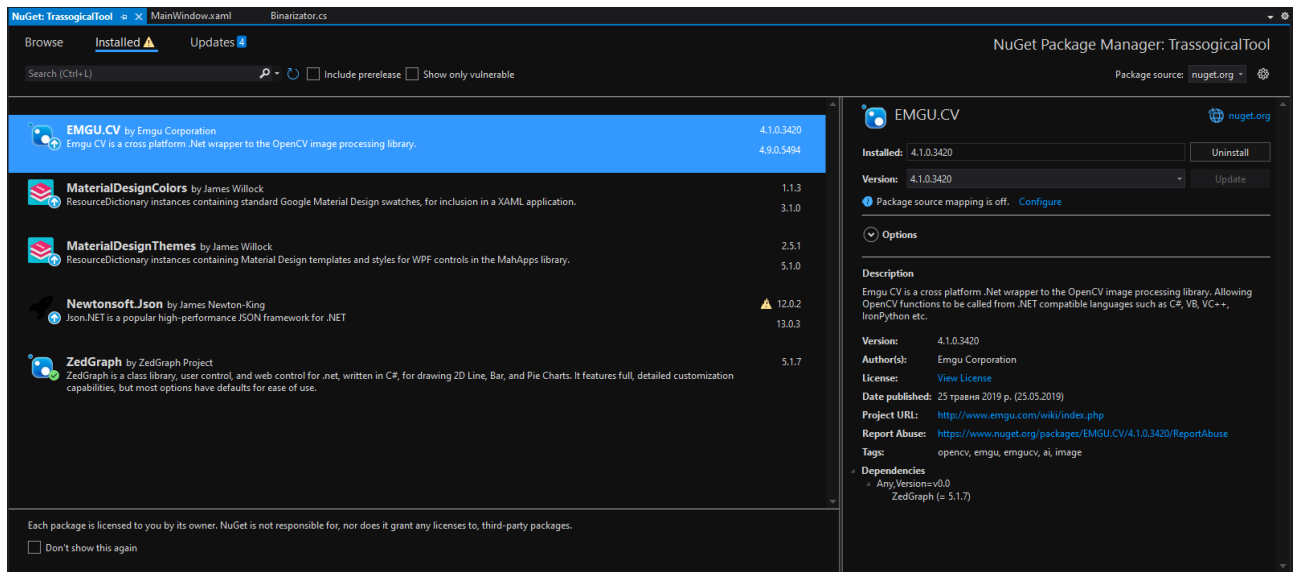


Рис. 3.6. Меню під'єднання бібліотеки EMGUCV

Для під'єднання бібліотеки потрібно зайти в менеджер управління пакетами та знайти потрібну нам бібліотеку та під'єднати її (див. рис. 3.6.).

Вона дозволить нам використовувати всі її інструменти для роботи з графічними об'єктами (функції, які дозволять нам значно простіше маніпулювати з графічними об'єктами).

3.5. Алгоритм попередньої обробки зображення

Для перевірки роботи застосунку потрібно спочатку створити базу з фотографіями слідів взуття. І вже тоді можна розпочати подальшу обробку цих фотографій. Алгоритм обробки зображень (див. рис. 3.7).

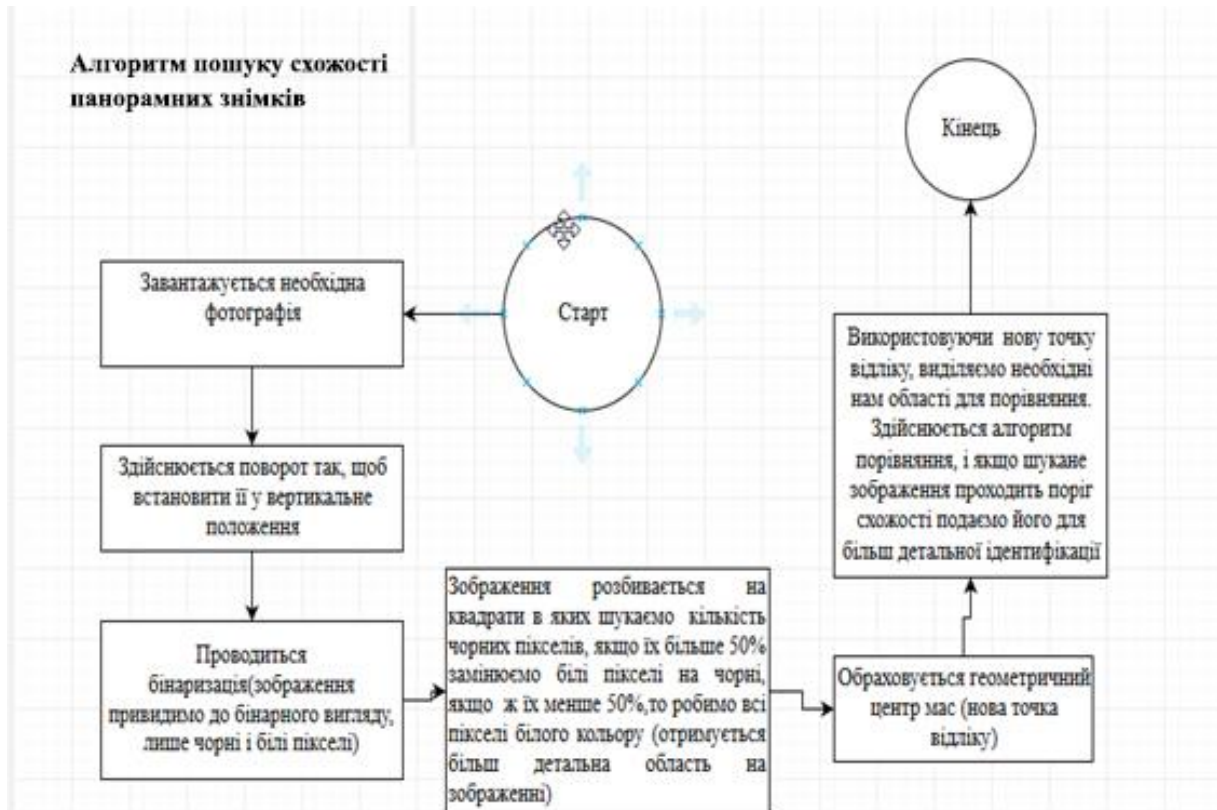


Рис. 3.7. Алгоритм обробки зображення

Опис кожного кроку алгоритму

1. Потрібно обрати зображення, яке ми плануємо обробляти.
2. Здійснюємо бінаризацію. Використовуючи функції бібліотеки EMGU.CV ми перетворюємо зображення до бінарного вигляду (чорні і білі пікселі), що дозволить нам в подальшому відкинути непотрібні нам області на фотографії і виділити ту область яка відповідає власне за відбиток взуття людини. Для ще кращого виділення області налаштовується контрастність зображення, яка коливається в межах від 0 до 100%.
3. Вважається, що всі панорамні знімки вносяться в базу у горизонтальному положенні, саме для цього здійснюється поворот уже бінаризованого зображення.
4. Знаходження геометричного центру мас потрібний нам для того щоб створити точку відліку для зображення, шукається він за формулою:

$$\left(X_c = \frac{1}{k} \sum_{i=0}^k X_i, Y_c = \frac{1}{k} \sum_{i=0}^k Y_i \right)$$

k – число чорних пікселів на зображенні, $\sum_{i=0}^k X_i$ – сума координат чорних пікселів по X , $\sum_{i=0}^k Y_i$ – сума координат чорних пікселів по Y .

Приклад роботи (див. рис. 3.8.) наведено нижче, де було знайдено і виділено червоним кольором центр мас для геометричної фігури трикутник на основі матриці з 0 (уявно білий колір) та 1 (уявно чорний колір).

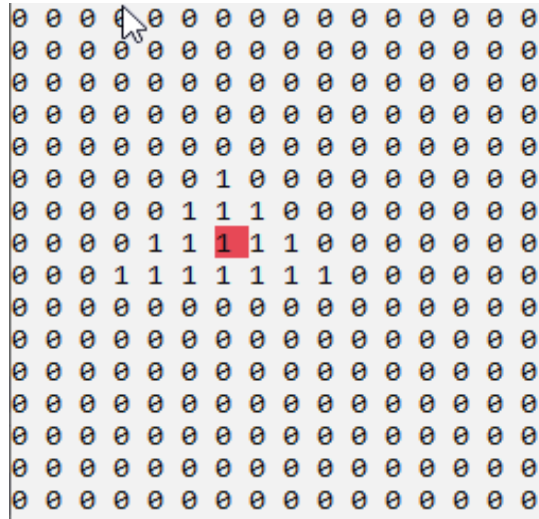


Рис. 3.8. Приклад пошуку центра мас

5. Перевірка схожості двох зображень.

Для початку визначаються межі за якими ми власне і будемо проводити перевірку, саме для цього ми шукали геометричні центри мас. Порівнюючи координати центру мас обох зображень ми знаходимо мінімальну відстань від центру до лівого, правого краю, до вершини і низу, тим самим визначаючи область для порівняння однакового розміру. Далі маніпулюючи циклами в C# здійснюється порівняння точок двох зображень, якщо вони співпадають то до лічильника додається 1.

Пройшовши всі точки, лічильник ділиться на загальне число точок і множиться на 100 для відображення у відсотковому співвідношенні, і якщо цей відсоток більший за пороговий, який ми встановили, це зображення додається у стек програми для подальшої обробки експертом.

Висновки до третього розділу

Розробка та використання програмного забезпечення є ключовим аспектом сучасної технологічної епохи, який значно спрощує взаємодію між людьми, технологіями та інформацією. У цьому контексті платформа Microsoft Visual Studio стає важливим інструментом для створення різноманітних типів програмного забезпечення, від настільних додатків до веб-застосунків і мобільних додатків. Завдяки її потужним можливостям і широкому спектру підтримуваних технологій, Visual Studio дозволяє розробникам створювати високоефективні рішення, які відповідають сучасним вимогам користувачів.

Технологія WPF (Windows Presentation Foundation) забезпечує розробників інструментарієм для створення багатофункціональних графічних інтерфейсів користувача з використанням сучасних графічних технологій. Основна перевага WPF — це можливість застосування апаратного прискорення через DirectX, що дозволяє створювати графічно складні додатки з високою продуктивністю. Використання декларативної мови розмітки XAML спрощує процес розробки інтерфейсів, дозволяючи розробникам чітко відокремити логіку бізнес-процесів від дизайну. WPF також надає можливості для створення адаптивних інтерфейсів, мультимедійних додатків і застосувань з тривимірною графікою.

Технологія JSON відіграє важливу роль у зберіганні та обміні даними, особливо коли мова йде про взаємодію між різними частинами програмного забезпечення. Використання JSON забезпечує ефективне збереження оброблених зображень і підтримує зворотну сумісність з багатьма типами даних. Це дозволяє легко інтегруватися з різними сервісами та технологіями, що робить JSON зручним форматом для обміну даними в сучасних веб-застосунках.

Процес інсталяції та налаштування Microsoft Visual Studio є відносно простим і зрозумілим, що робить його доступним для широкого кола розробників, включаючи новачків. Безкоштовна версія Visual Studio надає значні можливості для навчання та тестування різних функцій, що допомагає

розробникам адаптуватися до середовища і виявити необхідні інструменти для своєї роботи. Завдяки підтримці різних платформ і мов програмування, Visual Studio залишається універсальним інструментом, який відповідає всім вимогам сучасної розробки програмного забезпечення.

Використання сучасних інструментів і технологій, таких як Visual Studio, WPF та JSON, значно підвищує ефективність розробки програмного забезпечення, дозволяючи створювати якісні продукти з високими показниками продуктивності та зручності використання. Застосування цих технологій у практиці розробки забезпечує широкі можливості для створення інноваційних рішень, що відповідають найвищим стандартам якості в сучасній індустрії програмного забезпечення.

РОЗДІЛ 4

ІНСТРУКЦІЯ КОРИСТУВАЧА

При запуску програми відкривається її меню (див. рис. 4.1).

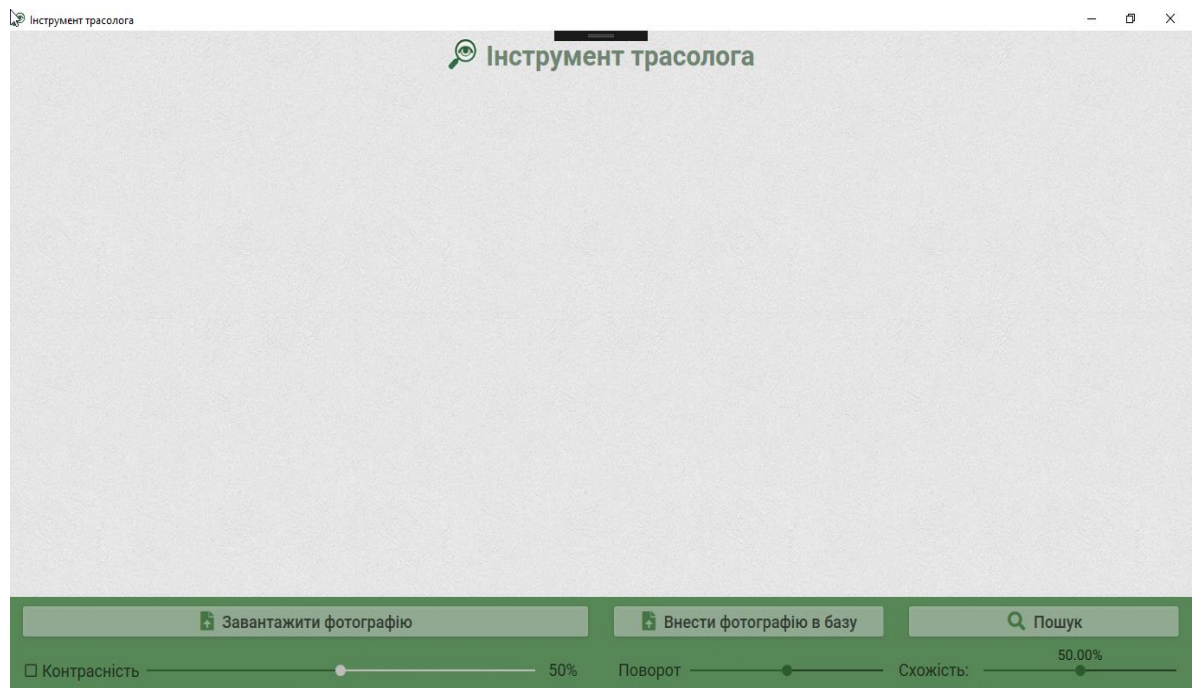


Рис. 4.1. Головне меню програми

- Для вибору фотографії потрібно натиснути кнопку «Завантажити фотографію» (див. рисунок 4.2.);

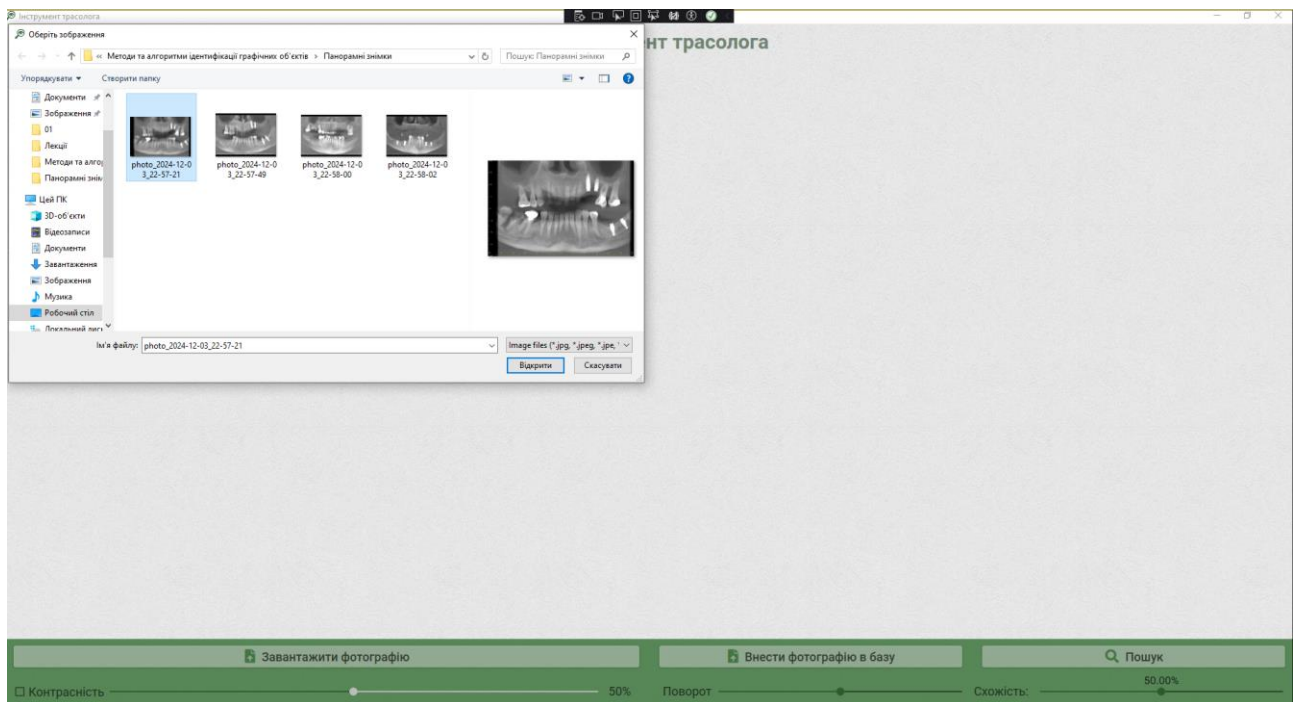


Рисунок 4.2. Вибір фотографії

- Щоб розпочати процес встановлення контрастності потрібно поставити галочку в CheckBox біля поля «Контрастність» (див. рисунок 4.3.), і відрегулювати рівень контрастності повзунком (див. рисунок 4.4.);

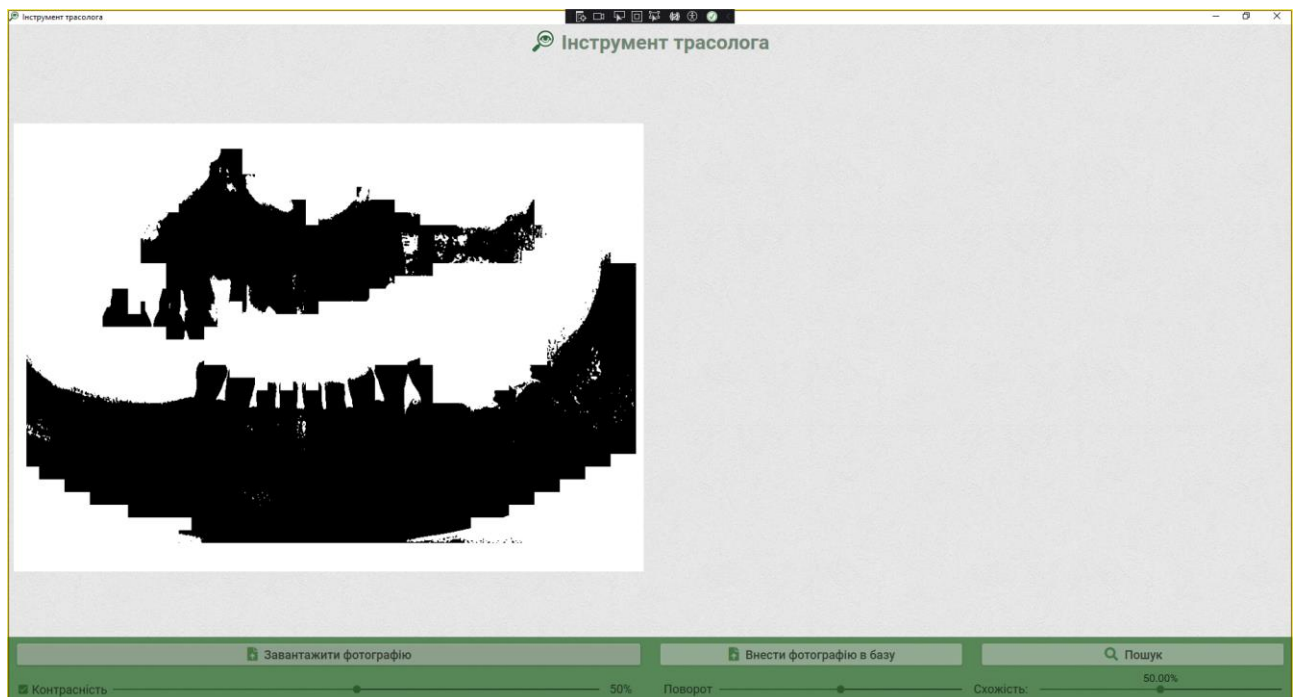


Рисунок 4.3. Регулювання контрастності



Рисунок 4.4. Відрегульована контрастність

- Якщо потрібно, то здійснити поворот повзунком, для встановлення зображення у вертикальне положення;
- Для внесення зображення у базу даних натиснути кнопку «Внести зображення в базу»(див. рисунок 4.5.);

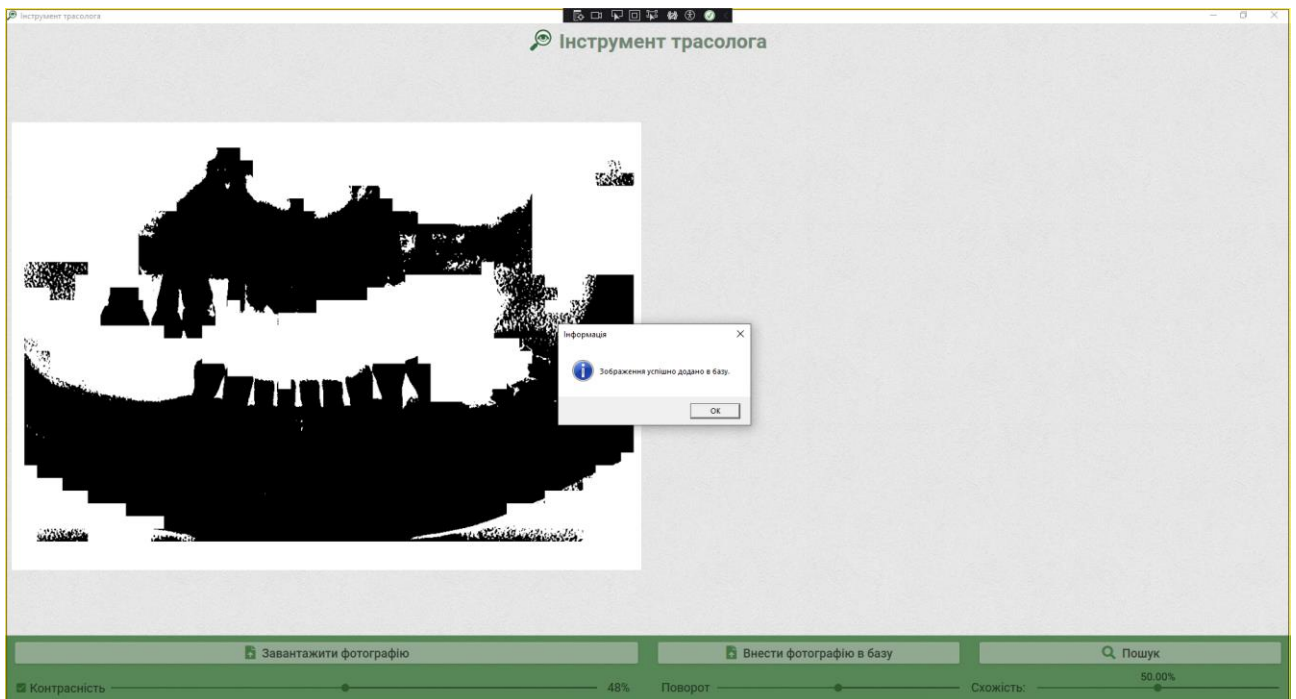


Рисунок 4.5. Успішне додавання фото в базу

- Для здійснення пошуку схожих зображень встановити повзунком мінімальний рівень схожості у відсотках і натиснути кнопку «Пошук».
- Для перегляду зображень, які підійшли нам за рівнем схожості гортаємо список фотографій справа.

Результат роботи програми має наступний вигляд (див. рис. 4.6.).

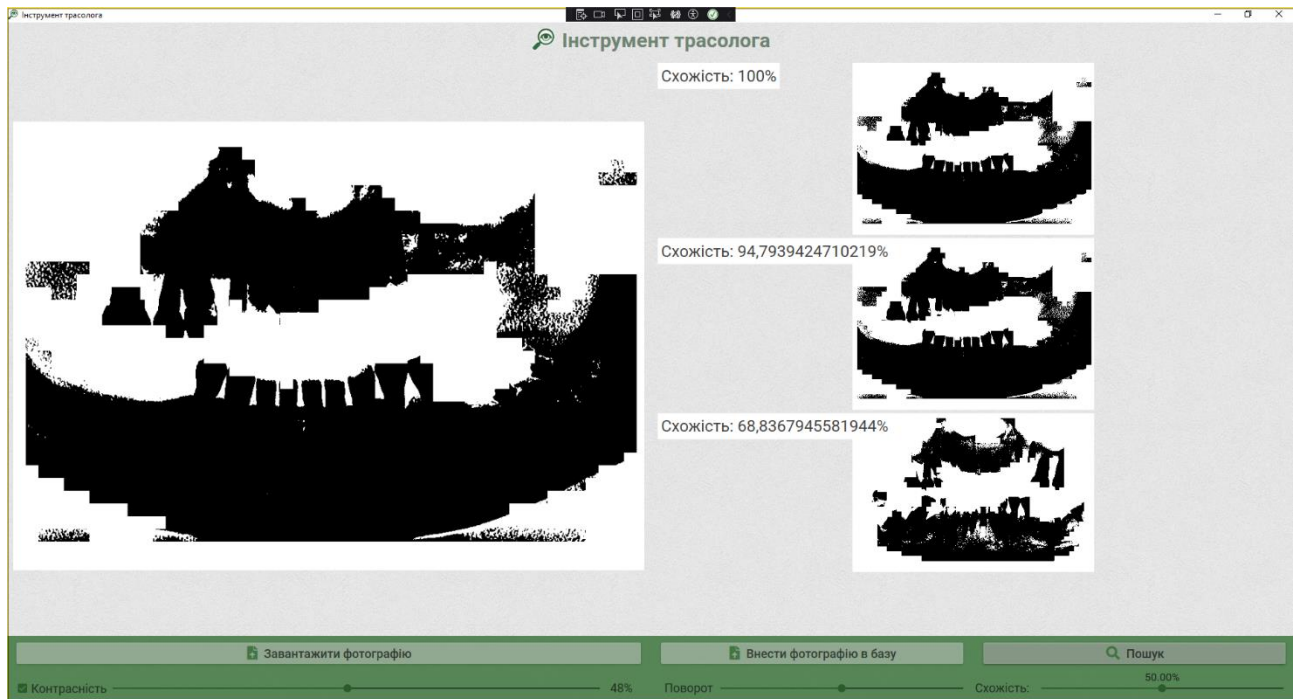


Рис. 4.6. Кінцевий вигляд роботи програми

Висновки до четвертого розділу

У четвертому розділі було розроблено та представлено детальну інструкцію користувача для роботи з програмним забезпеченням, створеним у рамках проведеного дослідження. Інструкція охоплює ключові етапи взаємодії із програмою, починаючи від завантаження фотографій і налаштування параметрів зображення до виконання пошуку схожих графічних об'єктів та перегляду результатів.

Особливу увагу приділено забезпеченню інтуїтивності інтерфейсу та зручності виконання основних операцій. Описано механізми налаштування

контрастності за допомогою повзунка, який дозволяє оптимізувати візуальне сприйняття зображення. Додатково охарактеризовано функцію повороту для корекції орієнтації графічного об'єкта, що особливо важливо для правильного внесення даних у базу.

Значна частина інструкції присвячена процесу збереження зображень у базі даних та налаштуванню параметрів пошуку схожості. Це включає можливість регулювання мінімального рівня відповідності у відсотках, що забезпечує високий рівень точності при ідентифікації графічних об'єктів. Функція перегляду результатів пошуку реалізована у зручній формі списку, що дозволяє швидко аналізувати знайдені об'єкти.

Візуальна ілюстрація ключових етапів роботи з програмою сприяє підвищенню доступності інструкції для користувачів. Наявність графічних матеріалів, таких як зображення інтерфейсу, значно спрощує процес освоєння програмного забезпечення навіть для користувачів із мінімальним технічним досвідом.

Інструкція спрямована на забезпечення ефективної роботи з програмою, підвищення її функціональності та полегшення інтеграції в робочі процеси. Розроблений матеріал сприяє оптимізації роботи з базою даних графічних об'єктів, забезпечуючи високу швидкість і точність виконання завдань.

ВИСНОВКИ

У межах магістерської роботи проведено глибоке дослідження та реалізовано програмне рішення для ідентифікації графічних об'єктів, що має практичне застосування в стоматології. Використання сучасних алгоритмів комп'ютерного зору у поєднанні з функціональними інструментами аналізу зображень забезпечило створення системи, здатної ефективно виконувати завдання з розпізнавання, обробки та класифікації графічних даних. Це дозволяє вирішувати складні задачі, зокрема, ідентифікацію стоматологічних зображень із високою точністю та оперативністю.

Розроблений програмний продукт демонструє високу ефективність при виконанні поставлених завдань, завдяки інтеграції адаптивного графічного інтерфейсу, алгоритмів обробки зображень і баз даних. Це забезпечує користувачам зручність у роботі з програмою, можливість гнучкого налаштування параметрів, таких як рівень контрастності, орієнтація зображень, а також проведення точного пошуку за заданими критеріями схожості. Такий підхід дозволяє суттєво підвищити якість аналізу стоматологічних зображень, зменшуючи ймовірність діагностичних помилок і покращуючи якість прийнятих рішень.

Впроваджене рішення має практичну значущість для медичної сфери, адже забезпечує автоматизацію рутинних процесів, оптимізацію часу та ресурсів у клінічній практиці. Система дозволяє не лише ефективно обробляти та зберігати дані, але й використовувати їх для подальшого аналізу та прийняття рішень, що є важливим для вдосконалення діагностичних процесів.

Окрім безпосереднього застосування в стоматології, створене програмне забезпечення демонструє великий потенціал для впровадження у суміжних галузях медицини та інших сферах, які потребують автоматизації обробки графічних даних. Додаткові перспективи включають інтеграцію з іншими медичними інформаційними системами, розширення функціоналу, наприклад, шляхом додавання можливостей тривимірного аналізу зображень або

застосування алгоритмів штучного інтелекту для покращення точності ідентифікації.

Таким чином, виконана робота стала важливим внеском у розвиток технологій обробки медичних зображень, демонструючи перспективність використання сучасних інформаційних технологій для вирішення реальних проблем. Отримані результати та напрацьовані методики створюють підґрунтя для подальших досліджень і розробок, спрямованих на вдосконалення методів ідентифікації графічних об'єктів і розширення їхнього практичного застосування.

Перспективою модифікації роботи є можливість застосовувати:

- оптимізацію алгоритмів ідентифікації для покращення продуктивності та точності;
- адаптацію алгоритмів для роботи зі складними випадками, наприклад, з пошкодженими або низькоякісними знімками;
- розширення функціональності програми, зокрема інтеграцію з базами даних пацієнтів або підтримку додаткових форматів зображень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bishop, C. M. Pattern Recognition and Machine Learning. Springer, 2006. – 738 p.
2. Murphy, K. P. Machine Learning: A Probabilistic Perspective. MIT Press, 2012. – 1104 p.
3. Duda, R. O., Hart, P. E., & Stork, D. G. Pattern Classification. 2nd ed. Wiley, 2000. – 680 p.
4. Goodfellow I., Bengio Y., Courville A. Deep Learning. – Cambridge: MIT Press, 2016. – 775 p.
5. Vasilev I., Slater D., Spacagna G., Roelants P., Baroukh V. Python Deep Learning. – Birmingham: Packt Publishing, 2024. – 528 p.
6. Pereverzev S.V., Smal A., & Tikhanov V. Algebraic Methods in Machine Learning. Springer, 2023. – 412 p. ISBN: 9783031298752.
7. Modern Algebraic Approaches to Pattern Recognition. Edited by Anderson L., Smith J., and Klein H. Academic Press, 2024. – 350 p. ISBN: 9780128195520.
8. Обробка зображень. – Режим доступу: https://uk.wikipedia.org/wiki/Обробка_зображень
9. EMGUCV. – Режим доступу: http://www.emgu.com/wiki/index.php/Main_Page
10. Постолит А.В. Visual Studio .NET: разработка приложений баз данных / А.В. Постолит – СПб.: БХВ-Петербург, 2003. – 544 с.
11. Введение в C#. – Режим доступу: <https://metanit.com/sharp/tutorial/1.1.php>
12. Интегрированная среда разработки Visual Studio. – Режим доступу: <https://msdn.microsoft.com/ru-ru/library/dn762121.aspx>
13. EMGUCV на C#. – Режим доступу: <http://qaru.site/questions/858540/how-to-install-and-use-emgu-cv-in-c>
14. Основы технологий WPF. Режим доступу: <https://metanit.com/sharp/wpf/>
15. Введение в JSON. – Режим доступу: <https://metanit.com/sharp/mvc/10.7.php>

16. Введение в ADO.NET. – Режим доступу: <https://metanit.com/sharp/ado-net/1.1.php>
17. Прошина М. Ю. Розпізнавання образів [Електронний ресурс] / М. Ю. Прошина. – 2012. – Режим доступу до ресурсу: https://wiki.tntu.edu.ua/Розпізнавання_образів.
18. Копча-Горячкіна Г. Е. ТЕОРІЯ РОЗПІЗНАВАННЯ ОБРАЗІВ / Г. Е. Копча-Горячкіна // Навчально-методичний посібник. Частина I для студентів факультету інформаційних технологій напрямку „Комп’ютерні науки” та „Програмна інженерія” ДВНЗ «Ужгородський національний університет» / Галина Ернестівна Копча-Горячкіна. – Ужгород: Видавництво ДВНЗ «Ужгородського національного університету», 2016. – С. 2–57.
19. Теорія розпізнавання образів. [Електронний ресурс] – Режим доступу до ресурсу: https://sites.google.com/site/navcalnijsajtdlastudentivvnz/teoreticni-osnoviinformatiki/laboratorni-roboti/lab_6.
20. Довбиш А. С. ОСНОВИ ТЕОРІЇ РОЗПІЗНАВАННЯ ОБРАЗІВ / А. С. Довбиш, І. В. Шелехов – Суми: Сумський державний університет, 2015. 109 с.