МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ

(підпис)

_____ червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,

освітньо-професійної програми «Інформатика»

на тему: «Інваріантна до рівня освітлення інформаційна система розпізнавання фрагментів зображень»

здобувача групи ІН - 93 Винник Катерини Євгенівни

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Катерина Винник

(підпис)

Керівник,
Доцент кафедри Віктор Авраменко _____

(підпис)

Консультант,
Старший викладач кафедри
іноземних мов та лінгводидактики Ніна Мальована _____

(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»
В.о. завідувача кафедри
_____ Ігор ШЕЛЕХОВ
(підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН-93 Винник Катерини Євгенівни

1. Тема роботи: «Інваріантна до рівня освітлення інформаційна система розпізнавання фрагментів зображень» затверджую наказом по СумДУ від __
2. Термін здачі здобувачем кваліфікаційної роботи *до 09 червня 2023 року*
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) *1) Аналіз джерел за темою та літературний огляд матеріалу . 2) Огляд технологій, що використовуються. 3) Розробка алгоритму та програмного коду.*
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)_____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «____» _____ 20 ___ р.

Завдання прийняв до виконання _____ (підпис) Керівник _____ (підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	*Опис проблеми інваріантності до рівня освітлення в системах розпізнавання фрагментів зображень. Визначення мети та об'єкта дослідження.*	20.05.2023	
2	*Визначення та обґрунтування методології, яка буде використовуватися для розробки інформаційної системи.*	26.05.2023	
3	*Розробка алгоритмів та програмного забезпечення для розпізнавання фрагментів зображень з урахуванням інваріантності до рівня освітлення. Реалізація та налаштування системи.*	30.05.2023	
5	*Формулювання висновків.*	05.06.2023	

Здобувач вищої освіти _____ (підпис) Керівник _____ (підпис)

АНОТАЦІЯ

Записка: 68 ст., 11 рис., 1 табл., 20 використане джерело.

Обґрунтування актуальності теми роботи – У сучасному світі інформаційні системи розпізнавання фрагментів зображень мають широке застосування в багатьох галузях, таких як комп'ютерне зорове сприйняття, медична діагностика, безпека та багато інших. Проте, багато з цих систем не є інваріантними до рівня освітлення, що може призводити до зниження точності та надійності їх роботи. Тому актуальним є дослідження та розробка інваріантної до рівня освітлення інформаційної системи розпізнавання фрагментів зображень.

Об'єкт дослідження – інформаційна система розпізнавання фрагментів зображень.

Мета роботи – розробка інваріантної до рівня освітлення інформаційної системи розпізнавання фрагментів зображень, яка здатна забезпечити стабільність та точність розпізнавання незалежно від умов освітлення.

Методи дослідження – Аналіз літературних джерел для отримання необхідних знань про існуючі методи розпізнавання фрагментів зображень і їх інваріантність до освітлення. Розробка математичної моделі. Реалізація розробленої моделі та виконання експериментів для оцінки її можливостей.

Результат – Розроблена інваріантна до рівня освітлення інформаційна система розпізнавання фрагментів зображень. Проведений аналіз та оцінка ефективності розробленої системи.

ЗМІСТ

1. ВСТУП

1.1 Актуальність теми

У сучасному світі, де використання комп'ютерного зору і розпізнавання образів стає все більш поширеним, проблема інваріантності до рівня освітлення при розпізнаванні фрагментів зображень є актуальною. Освітлення є одним з найважливіших факторів, що впливають на зображення, і його нерегулярності можуть викликати значні труднощі при розпізнаванні об'єктів або образів на зображеннях. Розробка інформаційної системи, яка була б інваріантною до рівня освітлення, має великий потенціал у багатьох галузях, таких як медицина, безпека, автомобільна промисловість та багато інших.

1.2 Мета дослідження

Основною метою дослідження є розробка інформаційної системи розпізнавання фрагментів зображень, яка повинна бути інваріантною до рівня освітлення. Для цього треба розробити методи та алгоритми, що дозволять системі ефективно розпізнавати фрагменти зображень незалежно від освітлення. Основні завдання включають аналіз існуючих методів, розробку нових алгоритмів та методів просторової нормалізації, а також оцінку ефективності запропонованих рішень.

1.3 Завдання дослідження

В рамках дослідження планується виконати такі завдання:

- Аналіз сучасних методів розпізнавання фрагментів зображень та їх інваріантності до рівня освітлення.
- Вивчення впливу освітлення на процес розпізнавання.

- Реалізація інформаційної системи розпізнавання фрагментів зображень, використовуючи запропоновані алгоритми та методи.

- Проведення експериментів з використанням розробленої системи та оцінка її ефективності на різних тестових наборах даних.

1.4 Предмет дослідження

Предметом дослідження є інформаційна система розпізнавання фрагментів зображень, яка забезпечує інваріантність до рівня освітлення. Дослідження спрямоване на розробку нових методів і алгоритмів, які дозволять системі ефективно розпізнавати фрагменти зображень незалежно від освітлення.

2. ОГЛЯД ЛІТЕРАТУРИ

2.1Основні відомості з теми

Інваріантна до рівня освітлення інформаційна система розпізнавання фрагментів зображень підвищує надійність розпізнавання фрагментів незалежно від умов освітлення.

З цією метою використовуються різні методи та підходи. Наприклад, одним з підходів є використання методів нормалізації освітлення, які дозволяють зменшити вплив змін освітлення на зображеннях [1]. Це може включати адаптивну регуляцію контрастності [2], видалення тіней або використання спеціальних фільтрів [3].

Також досліджується використання інваріантних ознак для розпізнавання фрагментів. Інваріантні ознаки не залежать від рівня освітлення і дозволяють точно визначати об'єкти або патерни на зображеннях. Це можуть бути

геометричні ознаки, текстурні ознаки або інші характеристики, які залишаються стійкими навіть при зміні умов освітлення [4].

Важливим аспектом є також аналіз попередніх досліджень у галузі розпізнавання зображень, зокрема методів розпізнавання фрагментів зображення [5]. Це дає змогу виявити наявні методи, їх переваги та недоліки, а також ідентифікувати прогалини, які можуть бути заповнені власним дослідженням.

Враховуючи ці основні відомості, необхідно продовжити розвиток інформаційних систем розпізнавання фрагментів зображень, які інваріантні до рівня освітлення. При цьому необхідно використовувати нові алгоритми, методи нормалізації та інші інваріантні ознаки для досягнення більш точного та надійного розпізнавання [4,6].

2.2 Аналіз попередніх досліджень у галузі розпізнавання зображень

Інваріантна до рівня освітлення інформаційна система з використанням функцій непропорційності була приведена у дослідженні [7]. У цьому дослідженні автори запропонували новий підхід до розпізнавання фрагментів зображень, який базується на використанні функцій непропорційності. Система враховує зміни у рівні освітлення шляхом аналізу функцій непропорційності для отриманих зображень. Результати дослідження підтверджують високу стійкість до змін освітлення та точність розпізнавання фрагментів зображень.

Додатково, цій темі також присвячені роботи [8-11]. Експериментальні результати, представлені в цих роботах, підтверджують ефективність використання функцій непропорційності для розпізнавання фрагментів зображень при зміні освітлення.

З урахуванням цих досліджень можна зробити висновок, що використання функцій непропорційності в системах розпізнавання фрагментів зображень є

актуальним підходом для досягнення інваріантності до рівня освітлення. Цей підхід має потенціал застосування в різних галузях, таких як комп'ютерне зорове сприйняття, розпізнавання образів та автоматичне управління системами.

Одним із можливих варіантів такої системи із використанням функції непропорційності є інформаційна система, в якій використовуються функції непропорційності [12].

Функція непропорційності допомагає створити інваріантну до рівня освітлення інформаційну систему розпізнавання фрагментів зображень, яка може ефективно працювати навіть при значних змінах освітлення. Це важливо для багатьох застосувань, таких як розпізнавання облич, детекція об'єктів, медична діагностика та безпека [13].

2.3 Функції непропорційності

2.3.1 Непропорційність по похідній n-порядку

Відомі кілька типів функції непропорційності: непропорційність по похідній n-порядку, непропорційність по значенню n-порядку і відносні непропорційності. Всі вони являються характеристиками числових функцій і характеризують наявність або відсутність пропорціонального зв’язку між цими функціями.

Непропорційність по похідній n-порядку y(x) по x описуються наступним виразом

$$@d_x^{(n)} y = \frac{y}{x^n} - \frac{1}{n!} \cdot \frac{d^n y}{dx^n}. \qquad (1)$$

Тут використовується символ @ для позначення операції обчислення непропорційності. Символ «*d*» означає «похідна». Порядок вказується в круглих дужках. Ліва частина (1) звучить так: " *d n y* по *x*".

Непропорційність (1) дорівнює нулю, якщо функція y(*x*) має вигляд $y = kx^n$ і вона інваріантна по відношенню до коефіцієнта пропорційності *k*.

Непропорційність по похідній першого порядку (*n*=1) має вигляд:

$$@d_x^{(1)}y = \frac{y}{x} - \frac{dy}{dx}. \qquad (2)$$

Якщо функції $x = \varphi(t)$, $y = \psi(t)$, визначаються параметрично (*t* - параметр)

То непропорційність (2) має вигляд:

$$@d_{\phi(t)}^{(1)}\psi(t) = \frac{\psi(t)}{\phi(t)} - \frac{d\psi/dt}{d\phi/dx}. \qquad (3)$$

Якщо існує пропорційна залежність $\psi(t) = k\,\varphi(t)$ між двома функціями $\psi(t)$ і $\varphi(t)$, то непропорційність (3) дорівнює нулю у всій області їх існування, незалежно від значення коефіцієнта пропорційності.

Слід зазначити, що функції непропорційності дозволяють обчислювати невідомі коефіцієнти перед відомими функціями, що утворюють суму *y*(*x*), використовуючи значення, отримані для поточного значення аргументу.

Для випадку, коли функції не гладкі і не мають похідних, замість непропорційності (3) пропонується використовувати інтегральну непропорційність першого порядку [13, 14]. Ця непропорційність функції y(x) по відношенню до f(x) має вигляд:

$$@\mathrm{I}^{(1)}_{\mathrm{f(x)}}\mathrm{y(x)} = \frac{\int_{\mathrm{x-h}}^{\mathrm{x}} \mathrm{y(x)dx}}{\int_{\mathrm{x-h}}^{\mathrm{x}} \mathrm{f(x)dx}} - \frac{\mathrm{y(x)}}{\mathrm{f(x)}} \qquad (4)$$

де h - задане число.

У цьому випадку y(x) і f(x) представлені одновимірними масивами. Якщо апроксимувати інтеграли в (4) за допомогою формули трапецій з однаковим кроком h для y(x) і f(x), то непропорційність (4) набуває вигляду (5):

$$\mathrm{I_i} = @\mathrm{I}^{(1)}_{\mathrm{f_i}}\mathrm{y_i} = \frac{\mathrm{y_{i-1}} + \mathrm{y_i}}{\mathrm{f_{i-1}} + \mathrm{f_i}} - \frac{\mathrm{y_i}}{\mathrm{f_i}} \qquad (5)$$

Наступне зворотне перетворення також буде використано в даній роботі:

$$y_i = \frac{(y_{i-1} - I_i \cdot (f_{i-1} + f_i)) \cdot f_i}{f_{i-1}}. \qquad (6)$$

2.3.2 Властивості непропорційності по похідній n-порядку

Нижче наведені деякі властивості похідної диспропорційності:

1. Множення функції $y = f(x,t)$ на постійний множник C призводить до множення її непропорційності над похідною n-порядку.

2. Непропорційність по похідній n-порядку від суми (різниці) функцій дорівнює сумі (різниці) їх непропорційностей.

3. Для $y = k(t)x^n$, де n - ціле число більше нуля, з фіксованим значенням t $@d_x^{(n)} y_t = 0$всьому обсязі функції.

4. Ще одна властивість є наслідком теореми [12].

Дана функція $y = f(x,t)$, $x \in X,\ t \in T$. Тоді $@d_x^{(n)} y_t \neq 0$, але $@(n)@d_x^{(1)} y_t = 0$. Доведемо, що в даному випадку $y = f(x,t)$ має вигляд

$$y = k_n(t)x^n + k_{n-1}(t)x^{n-1} + \cdots + k_1(t)x, \qquad (9)$$

де $k_n(t),\ k_{n-1}(t),\ \ldots,\ k_1(t)$ - коефіцієнти, фіксовані для заданої величини t;

$\boldsymbol{k_n(t) \neq 0}$ і хоча б один з інших коефіцієнтів не дорівнює нулю; n - ціле число, більше нуля.

Доказ. Переконаємося, що непропорційність по похідній n-порядку не дорівнює нулю. Відповідно до

$$@d_x^{(n)} y_t = \frac{1}{x^n}[k_{n-1}(t)x^{n-1} + k_{n-2}(t)x^{n-2} + \cdots + k_1(t)x]. \qquad (10)$$

За умовою хоча б один з $k_{n-1}(t),\ k_{n-2},\ \ldots,\ k_1(t)$ не дорівнює нулю.

Тому непропорційність теж не дорівнює нулю. Тепер знайдіть послідовність n непропорційність над похідною першого порядку для $@(n)@d_x^{(1)} y_t$.

Для зручності кожна непропорційність в цій послідовності буде позначатися Z з відповідним індексом.

$$Z_1 = @d_x^{(1)} y_t = -[(n-1)k_n(t)x^{n-1} + (n-2)k_{n-1}(t)x^{n-2} + \cdots + 2k_3(t)x^2 + k_2(t)x];$$

$$Z_2 = @d_x^{(1)} Z_1 = [(n-1)(n-2)k_n(t)x^{n-2} + (n-2)(n-3)k_{n-1}(t)x^{n-3} + \cdots + 3 \cdot 2k_4(t)x^2 + 2k_3(t)x];$$

$$\vdots$$

$$Z_i = @d_x^{(1)} Z_{i-1} = (-1)^i \sum_{j=i}^{n} k_j(t) x^{j-i} \prod_{m=1}^{i} (j-m);$$

$$\vdots$$

$$Z_{n-1} = @d_x^{(1)} Z_{n-1} = (-1)^{n-1} (n-1)!\, k_n(t) x;$$

$$Z_n = @d_x^{(1)} Z_{n-1} = 0.$$

2.4 Проблема інваріантності до рівня освітлення

Проблема інваріантності до рівня освітлення виникає в багатьох задачах комп'ютерного зору та обробки зображень, де необхідно аналізувати зображення незалежно від його освітлення. Оскільки освітлення може суттєво змінюватись, наприклад, через різницю в джерелах світла, відображення поверхонь або інші фактори, інваріантність до освітлення є важливою властивістю для забезпечення стійкості та точності аналізу зображень [15 - 18].

Одним з підходів до вирішення проблеми інваріантності до рівня освітлення є використання інваріантних ознак або дескрипторів зображень. Ці ознаки повинні бути стійкими до зміни освітлення, що дозволяє їх використовувати для розпізнавання об'єктів або порівняння зображень незалежно від освітлення.

Наприклад, одним з популярних методів для отримання інваріантних ознак є метод SIFT (Scale-Invariant Feature Transform). Він базується на пошуку локальних ключових точок на зображенні та створенні дескрипторів, які є стійкими до масштабування, зміщення та обертання зображення, а також відносно стійкими до змін освітлення [18].

Інший підхід полягає в використанні методів нормалізації або попередньої обробки зображень для видалення ефектів освітлення. Наприклад, можуть

застосовуватись методи корекції кольору або конвертації в простір кольорів, які дозволяють зменшити вплив освітлення на зображення та забезпечити більшу стійкість до зміни освітлення.

Окрім цього, існують методи машинного навчання, які можуть навчитися розпізнавати об'єкти незалежно від рівня освітлення шляхом навчання на великій кількості зображень з різними рівнями освітлення та знаходження інваріантних закономірностей [12].

Усі ці підходи спрямовані на забезпечення інваріантності до рівня освітлення та покращення якості та стійкості аналізу зображень незалежно від змін освітлення [19, 20, 15].

3. ПОСТАНОВКА ЗАДАЧІ

3.1 Постановка задачі

Розробити алгоритм і комп’ютерну програму для моделювання роботи системи розпізнавання фрагментів зображень при довільному рівні освітлення.

3.2 Математична постановка задачі

Дано два еталонних растрових зображення, кожний із яких може бути представлений матрицею. В ній h рядків і w стовпчиків. Кожному елементу матриці відповідає піксель зображення із складовими кольорів Red, Green, Blue. При скануванні зображення відбувається прочитання пікселів зліва направо і зверху вниз. При цьому відбувається зчитування значень складових кольорів. Кожне із них може бути від 0 до 255.

Позначимо:

i=0,1,…,h-1- номер рядка;

j=0,…,w-1- номер стовпчика;

q= iw+j – номер поточного пікселя;

$r_1(q)$, $g_1(q)$, $b_1(q)$ – відповідно значення червоної, зеленої і блакитної складових першого зображення.

$r_2(q)$, $g_2(q)$, $b_2(q)$ – складові кольорів пік селів другого зображення.

Дано також комбіноване зображення, на якому можуть бути фрагменти першого, або другого зображень а також ще якісь випадкові зображення. Необхідно для любого із заданих еталонів розпізнавати його фрагменти на комбінованому зображенні при довільному рівні освітлення. Позначимо через M рівень освітлення цього зображення, яке може бути випадковим в інтервалі від 0 до 1, що відповідає переходу від повної темноти до 100% освітлення. При зниженні рівня освітлення, пропорційно зменшуються значення складових кольору. Позначимо складові кольорів пікселів комбінованого зображення при довільному рівні освітлення через rr(q), gg(q), bb(q).

Для пікселів першого еталонного зображення

$rr(q) = Mr_1(q)$, (2)

$gg(q) = Mg_1(q)$, (3)

$bb(q) = Mb_1(q)$, (4)

Відповідно для другого:

$rr(q) = Mr_2(q)$, (5)

$gg(q) = Mg_2(q)$, (6)

$bb(q) = Mb_2(q)$, (7)

Тобто задача зводиться до виявлення пропорціонального зв’язку між складовими кольорів пікселів комбінованого зображення та кожного із еталонів. Якщо для якогось із еталонів такий зв’язок виявлений, вважається, що відповідний піксель належить цьому еталону.

Якщо для жодного із еталонів пропорційний зв’язок не виявлений, це означає присутність ще якогось стороннього зображення.

Цю задачу не можна розв’язати простим діленням значень складової кольору затемненого зображення на таку ж складову відповідного пікселя одного із досліджуваних зображень. Справа в тому, що результат ділення невідомо із чим порівнювати внаслідок невідомого рівня освітлення і відсутності інформації про те, якому еталону належить поточний піксель.

4. АЛГОРИТМ РОЗВ’ЯЗАННЯ ЗАДАЧІ

4.1 Хід розв’язання задач

Присвоюємо першому еталонному зображенню індекс k = 0, а іншому k = 1. Обчислимо інтегральну непропорційність першого порядку складових кольорів rr(q), gg(q), bb(q) комбінованого затемненого зображення по відповідним складовим кожного еталону.

Позначимо непропорційність по червоній складовій поточного пікселя для k-го еталона через $D_r(k,q)$, для залежної складової $D_g(k,q)$ і для блакитної $D_b(k,q)$, де

q – номер пікселя

$$D_r(k,q) = \frac{rr(q-1)+rr(q)}{r(k,q-1)+r(k,q)} - \frac{rr(q)}{r(k,q)}$$

$$D_g(k,q) = \frac{gg(q-1)+gg(q)}{g(k,q-1)+g(k,q)} - \frac{gg(q)}{g(k,q)}$$

$$D_b(k,q) = \frac{bb(q-1)+bb(q)}{b(k,q-1)+b(k,q)} - \frac{rr(q)}{r(k,q)}$$

де k = 0,1.

Аналіз отриманих непропорційностей дозволяє розпізнати пропорціональний зв’язок між складовими кольорів пікселів затемненого зображення і еталонних зображень.

У випадку коли $D_r\ (k, q)$ або $D_g\ (k, q)$ або $D_b\ (k, q)$ дорівнює нулю, це означає, що q-й піксель відповідає відповідному пікселю еталонного зображенню. Якщо пропорційності нульові для обох коефіцієнтів, це означає, що в обох еталонах складові пікселів співпадають.

У випадку, коли непропорційності не дорівнюють нулю, це означає, що q-й піксель не відповідає ні першому ні другому еталонам.

4.2 Алгоритм програми

При написанні алгоритму застосовуються ідентифікатори, які приведені в таблиці.

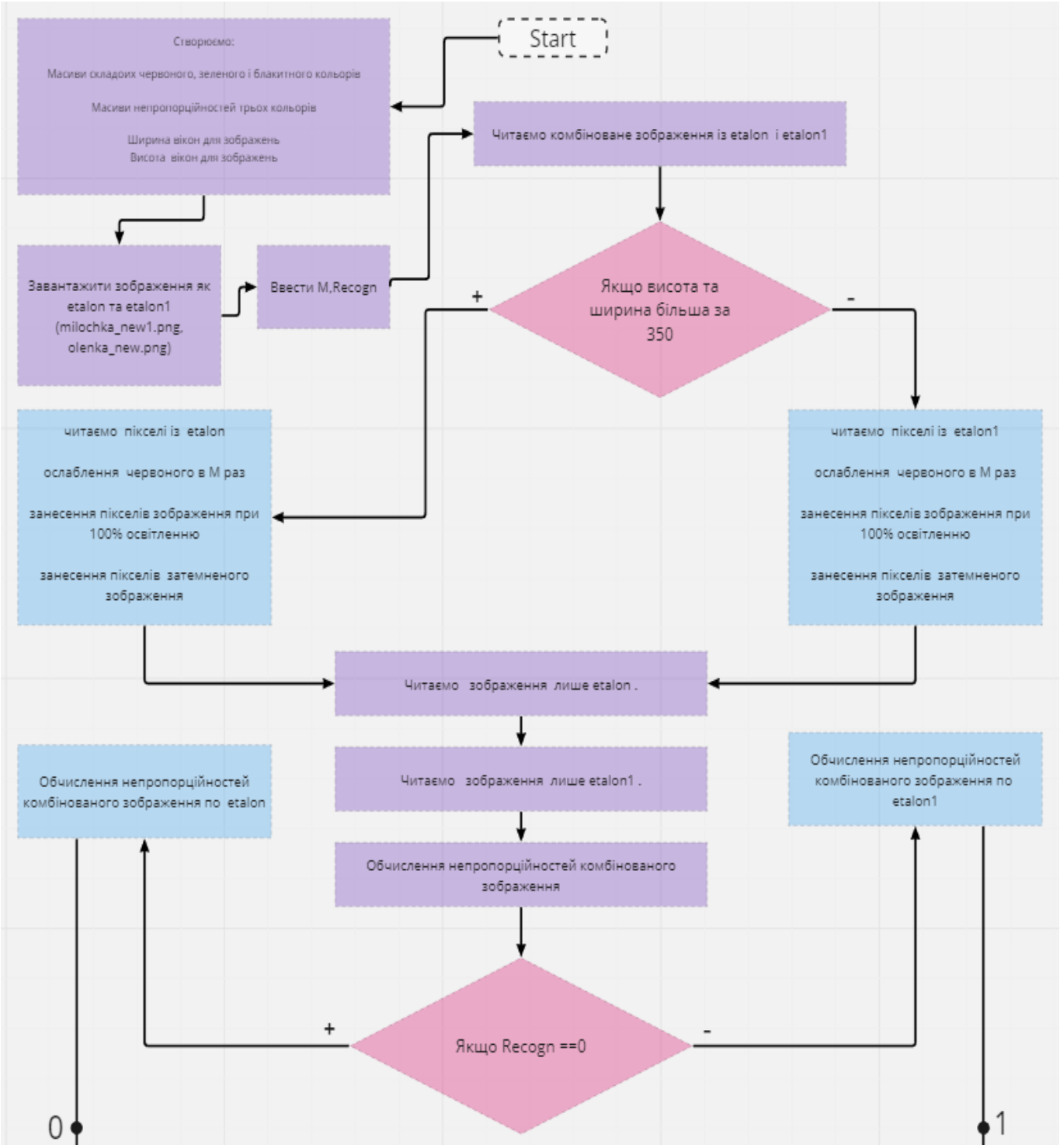
Таблиця 1.

Зміна	Ідентифікатор	Пояснення
w	width	Ширина вікна
h	height	Висота вікна
i,j	i,j	Номери рядка і стовпчика
q	q	Номер поточного пікселя

$r_1(q)$, $g_1(q)$, $b_1(q)$	R1[q], G1[q], B1[q]	Складові q-го пікселя 1 еталону
$r_2(q)$, $g_2(q)$, $b_2(q)$	R2[q], G2[q], B2[q]	Складові пікселя 2-го еталону
rr(q), gg(q), bb(q)	R_new_[q], G_new_[q], B_new_[q]	Складові пікселя комбінованого зображення
	etalon, etalon1	Значення еталонів
M	M	Рівень освітлення комбінованих зображень
k	Reogn	Номер зображення
$D_r(k,q)$ $D_g(k,q)$ $D_b(k,q)$	disp_r[q] disp_g[q] disp_b[q]	Непропорційність для пікселя еталону

	swr, swg, swb	Файли для значення складових

4.3 Блок-схема програми



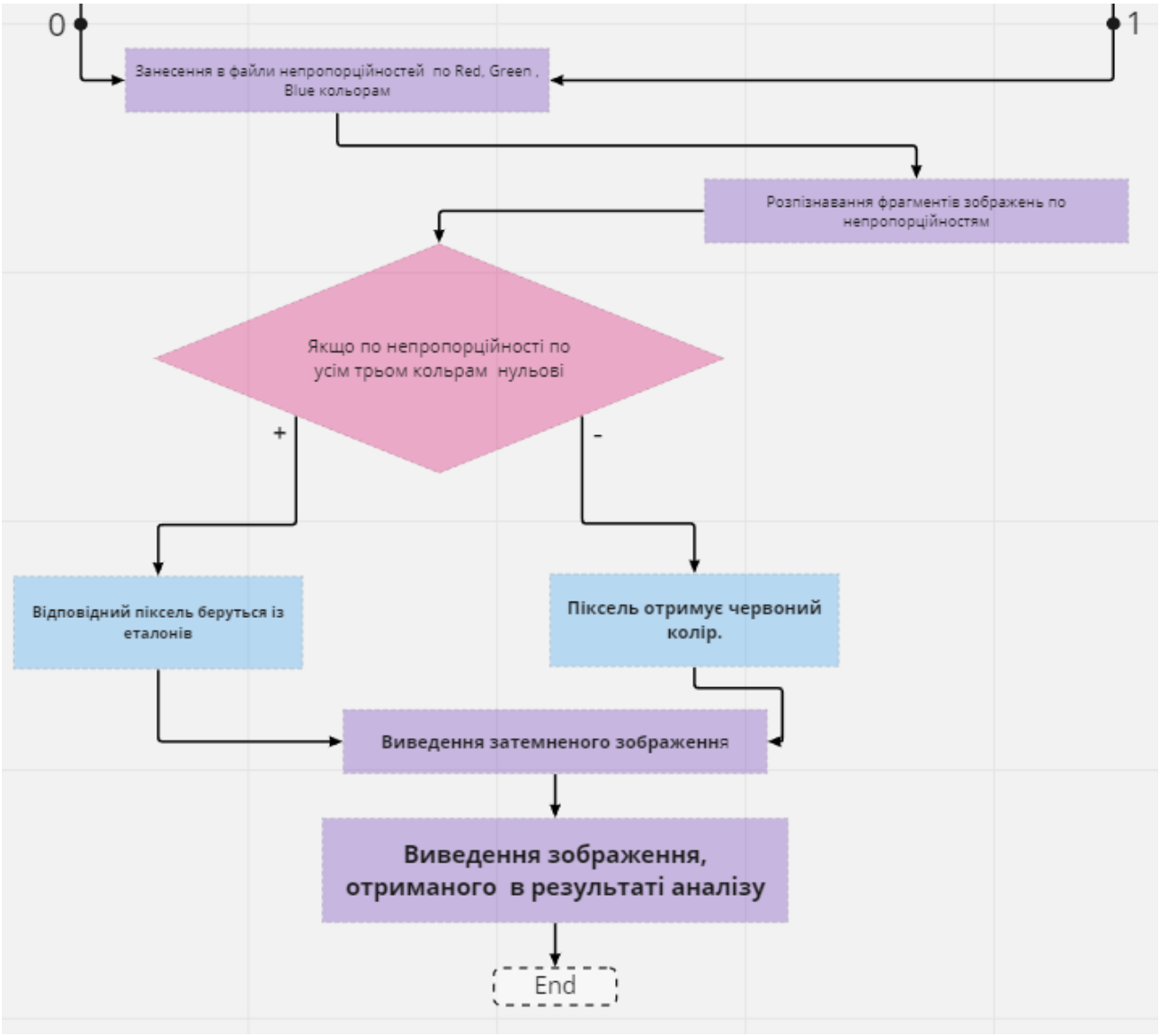


Рисунок 1. Блок схема що ілюструє роботу алгоритму програми.

4.4 Інструкція для користувача

Цей код на C# містить деякі функції та змінні, які виконують різні дії пов'язані з обробкою зображень.

У цьому коді використовуються різні простори імен, такі як `System`, `System.Collections.Generic`, `System.ComponentModel`, `System.Data`, `System.Drawing`, `System.Linq`, `System.Text`, `System.Threading.Tasks`, `System.Windows.Forms`, `System.IO` і `System.Reflection.Emit`. Кожен з них

надає доступ до певних класів та функцій, які потрібні для виконання певних завдань.

Код також визначає простір імен `WindowsFormsApp1` і клас `Form1`, який є початковою формою програми.

Основна логіка програми розміщена в методі `button1_Click`, який є обробником події для натискання кнопки з ідентифікатором `button1`. У цьому методі відбувається обробка зображень і розпізнавання фрагментів на основі непропорційності кольорів.

Код містить декілька масивів, таких як `r`, `g`, `b`, `r_new`, `g_new`, `b_new`, `R`, `G`, `B`, `R_new`, `G_new`, `B_new`, `R1`, `G1`, `B1`, `R2`, `G2`, `B2`, які використовуються для зберігання значень компонентів кольору пікселів.

Також у коді є функція `nonprop`, яка обчислює непропорційності для заданого масиву значень кольорів.

У цьому коді використовуються класи `Color`, `Graphics`, `Bitmap`, `StreamWriter`, які надають функціональність для роботи з кольорами, зображеннями, графікою та файлами.

Код також містить деякі операції зчитування та записування пікселів зображень, створення нових зображень, обчислення непропорційностей та розпізнавання фрагментів на основі непропорційностей.

Загалом, цей код виконує обробку зображень, розпізнавання фрагментів та збереження результатів в файл.

4.5 Середовище для розробки

Visual Studio 2022

Visual Studio є інтегрованим середовищем розробки (ICP) для створення програмного забезпечення, яке надає широкий спектр інструментів і можливостей. Це потужне ICP, яке дозволяє програмістам редагувати, налагоджувати, компілювати та публікувати код.

Visual Studio містить не тільки основний редактор і відладчик, але й компілятори, засоби автодоповнення коду, графічні конструктори та інші функціональні можливості, які сприяють поліпшенню процесу розробки програмного забезпечення.

Інтегроване середовище Visual Studio відоме своєю високою продуктивністю і швидкодією. Воно підтримує розробку на різних платформах та пристроях і забезпечує можливість компілювати програми різних типів. Крім того, Visual Studio надає зручні засоби спільної роботи в режимі реального часу, діагностику проблем та виправлення їх до виникнення.

Завдяки своїм функціональним можливостям, Visual Studio робить щоденні процеси розробки більш гнучкими та адаптивними, сприяючи підвищенню продуктивності розробників програмного забезпечення.

Створення проекту у Visual Studio на C#

1. Відкрити Visual Studio.

2. Вибрати "Створення проекту" або використати комбінацію клавіш Ctrl+Shift+N.

3. У вікні "Створення проекту" вибрати "Windows Forms Application" (додаток Windows Forms) в категорії "Visual C#".

4. Ввести ім'я проекту і вибрати папку для збереження.

5. Натиснути "OK", щоб створити проект.

6. Після створення проекту відкриється вікно "Form1.cs" (або інше ім'я, якщо ви вказали інше ім'я форми).

7. У вікні редактора форми можна додавати елементи керування (кнопки, текстові поля, списки і т.д.) з панелі "Toolbox".

8. Розмістити елементи керування на формі, клацнувши на них і перетягнувши на форму.

9. Щоб додати обробник подій для елемента керування (наприклад, обробник кліку на кнопці), необхідно подвійно клацнути на цьому елементі керуванні в редакторі форми. Це створить заготовку обробника подій і перенесе вас до редактора коду.

10. У редакторі коду всередині методу обробника треба написати код, який буде виконуватись при виникненні події (наприклад, при кліку на кнопку).

11. Після додавання елементів керування і написання коду зберегти файли проекту.

12. Запустити проект, натиснувши кнопку "Запустити" (або використати комбінацію клавіш Ctrl+F5).

13. Після запуску проекту з'явиться форма з елементами керування, і код буде виконуватись при взаємодії з формою.

Таким чином, ви можна створювати свої проекти на C# у Visual Studio з використанням форми, додавати необхідні елементи керування та писати відповідний код для обробки подій та логіки вашого додатку.

5. КОНТРОЛЬНІ ПРИКЛАДИ ТА РЕЗУЛЬТАТИ

5.1 Приклад 1 та його результати

Для тестування програми використано еталонні зображення двох кішок. Перша із них білого кольору з чорними п’ятанами-Алена, друга, сірого кольору –Міла. Ідентифікатор Міли Recogn [0], відповідно для Алени – Recogn [1].

Вхідні данні:

Перший експеримент проводився при рівні освітлення 10% M = 0.1; Та еталонні зображення Recogn = 0;

Результат:

Створення комбінованого зображення з урахуванням рівня освітленості M.

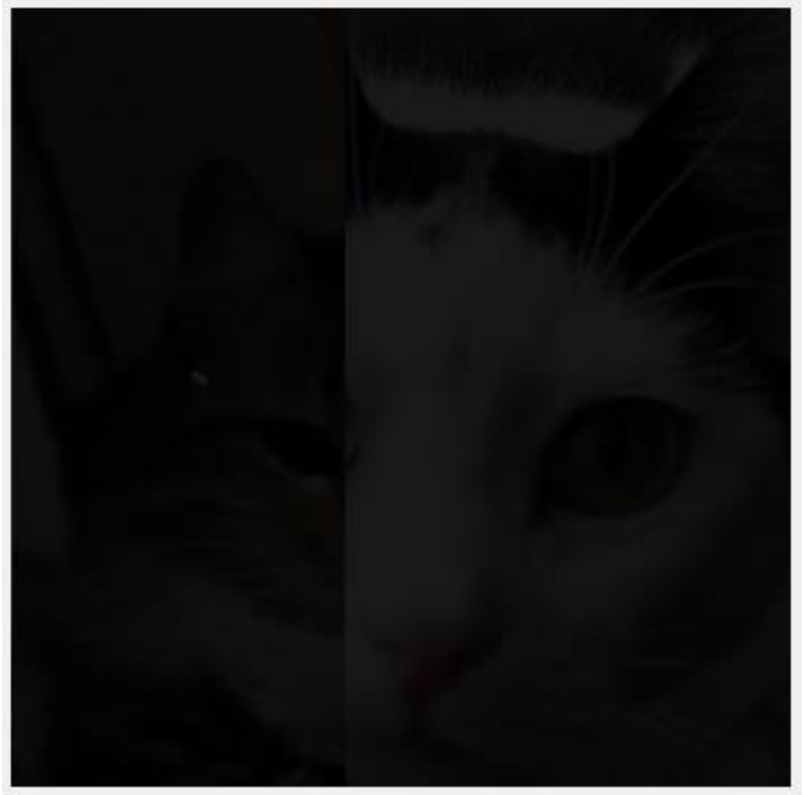


Рисунок 2. Комбіноване зображення з 10% яскравістю.

Розрахунок непропорційностей для кожного каналу кольору (червоний, зелений, синій) шляхом порівняння значень каналів для комбінованого зображення з відповідними значеннями для еталонного зображення (Recogn = 0).

Розпізнавання фрагментів зображення на основі непропорційностей. Якщо непропорційності для всіх трьох каналів кольору дорівнюють нулю, то відповідний піксель береться із еталонного зображення. В іншому випадку, піксель отримує червоний колір.

Створення зображення-результату, яке показує розпізнані фрагменти зображення.

Рисунок 3. Зображення Після обчислення непропорційностей комбінованого зображення по etalon (Recogn==0).

Результатом цього коду є запис непропорційностей кольорів у трьох окремих файли: "nonpr_red.txt" для червоного каналу, "nonpr_gr.txt" для зеленого каналу та "nonpr_bl.txt" для синього каналу.

Після виконання цього коду у вказаних файлах будуть міститися значення непропорційностей для червоного, зеленого і синього каналів кольору, які можна буде використовувати для подальшого аналізу чи обробки даних.

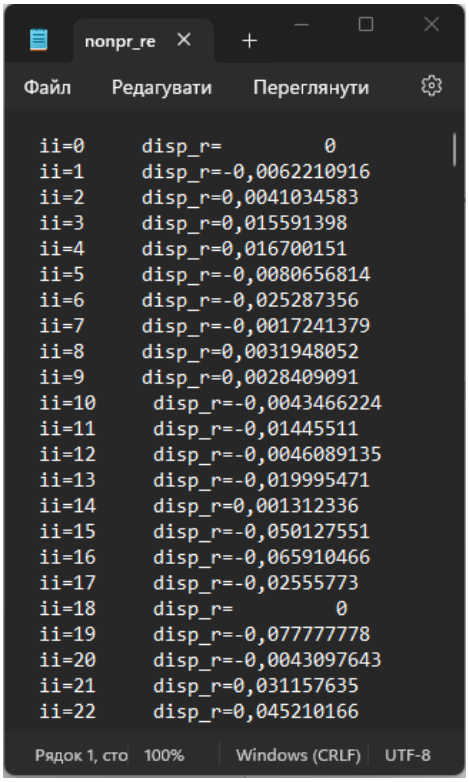


Рисунок 4. Результат занесення в файли непропорційностей Red, кольора.

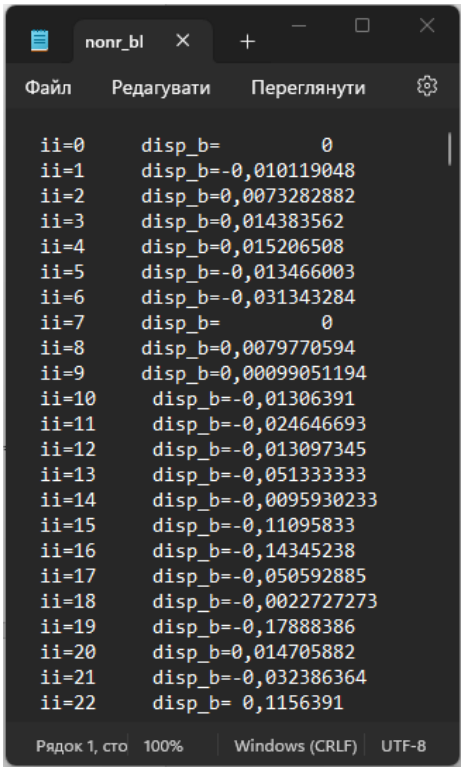


Рисунок 5. Результат занесення в файли непропорційностей Blue, кольора.

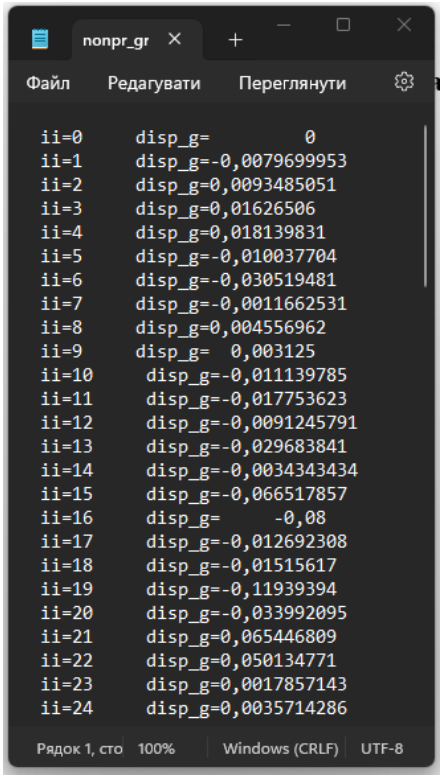


Рисунок 6. Результат занесення в файли непропорційностей Green, кольора.

5.2 Приклад 2 та його результати

Вхідні данні:

Перший експеримент проводився при рівні освітлення 10% M = 0.1; Та еталонні зображення Recogn = 1;

Результат:

Створення комбінованого зображення з урахуванням рівня освітленості M.

Рисунок 7. Комбіноване зображення з 10% яскравістю.

Розрахунок непропорційностей для кожного каналу кольору (червоний, зелений, синій) шляхом порівняння значень каналів для комбінованого зображення з відповідними значеннями для еталонного зображення (Recogn = 1).

Розпізнавання фрагментів зображення на основі непропорційностей. Якщо непропорційності для всіх трьох каналів кольору дорівнюють нулю, то відповідний піксель береться із еталонного зображення. В іншому випадку, піксель отримує червоний колір.

Створення зображення-результату, яке показує розпізнані фрагменти зображення.

Рисунок 8. Зображення Після обчислення непропорційностей комбінованого зображення по etalon1 (Recogn==1).

Результатом цього коду є запис непропорційностей кольорів у трьох окремих файли: "nonpr_red.txt" для червоного каналу, "nonpr_gr.txt" для зеленого каналу та "nonpr_bl.txt" для синього каналу.

Після виконання цього коду у вказаних файлах будуть міститися значення непропорційностей для червоного, зеленого і синього каналів кольору, які можна буде використовувати для подальшого аналізу чи обробки даних.

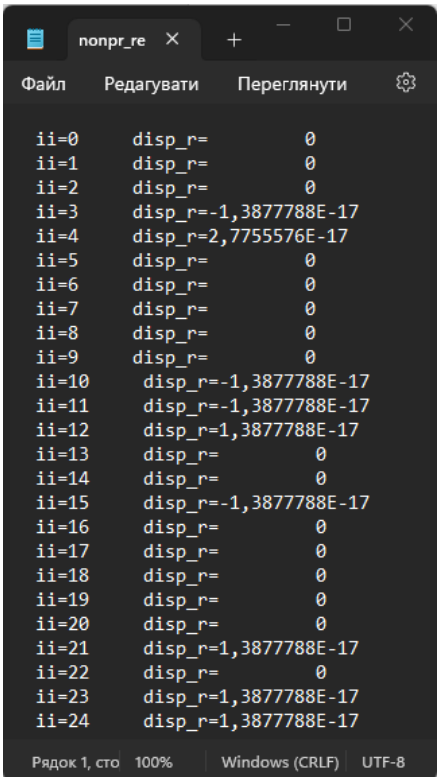


Рисунок 9. Результат занесення в файли непропорційностей Red, кольора.

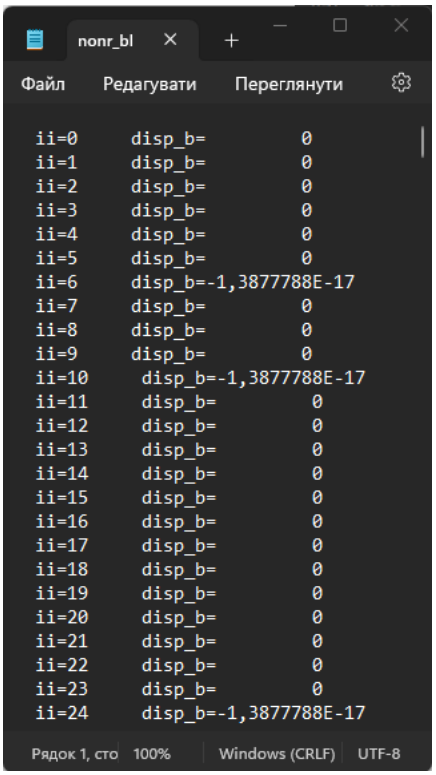


Рисунок 10. Результат занесення в файли непропорційностей Blue, кольора.

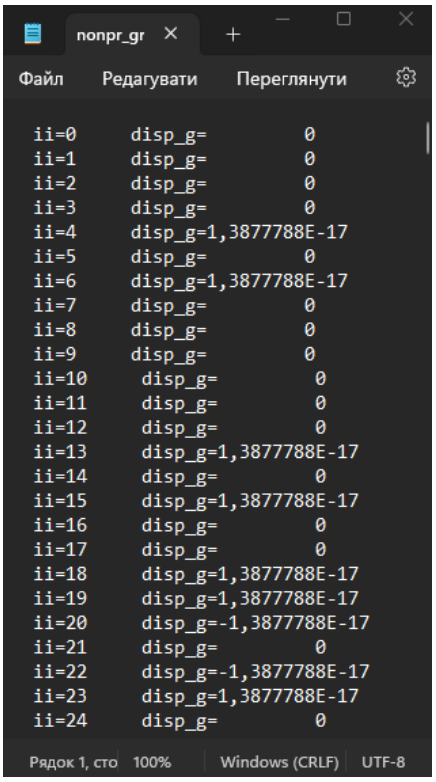


Рисунок 11. Результат занесення в файли непропорційностей Green, кольора.

ВИСНОВОК

В ході виконання кваліфікаційної роботи була розроблена інваріантна до рівня освітлення інформаційна система розпізнавання фрагментів зображень. Основними цілями роботи було досягнення стабільності та незалежності розпізнавання від рівня освітлення, що є актуальною проблемою в області комп'ютерного зору та обробки зображень.

Для досягнення поставлених цілей були проведені наступні кроки:

1. Проведений аналіз існуючих методів та підходів до розпізнавання зображень. Виявлені їх переваги та недоліки, зокрема, низьку стабільність до зміни освітлення.

2. Проведено експериментальне дослідження розробленої системи на наборі зображень з різним рівнем освітлення. Отримані результати підтвердили ефективність запропонованого підходу та його здатність до стабільного розпізнавання незалежно від освітлення.

Отже, можна зробити висновок, що розроблена інформаційна система розпізнавання фрагментів зображень є інваріантною до рівня освітлення. Ця система має великий потенціал для застосування в різних областях, де стабільність та незалежність від рівня освітлення є важливими факторами для ефективного розпізнавання зображень

ВАРІАНТ КВАЛІФІКАЦІЙНОЇ РОБОТИ АНГЛІЙСКОЮ МОВОЮ

1. Introduction

1.1 Relevance of the topic

In the modern world, where the use of computer vision and image recognition is becoming increasingly widespread, the problem of invariance to lighting conditions in image fragment recognition is relevant. Lighting is one of the most important factors that affect images, and its irregularities can pose significant challenges in recognizing objects or patterns in images. The development of an information system that is invariant to lighting conditions has great potential in various fields such as medicine, security, automotive industry, and many others.

1.2 Research objective

The main objective of this research is to develop an information system for the recognition of image fragments that is invariant to lighting conditions. To achieve this, methods and algorithms need to be developed that allow the system to effectively recognize image fragments regardless of the lighting conditions. The main tasks include analyzing existing methods, developing new algorithms and methods for spatial normalization, and evaluating the effectiveness of the proposed solutions.

1.3 Research tasks

Within the scope of this research, the following tasks are planned to be accomplished:

- Analyzing current methods of image fragment recognition and their invariance to lighting conditions.
- Studying the influence of lighting on the recognition process.

- Implementing an information system for the recognition of image fragments using the proposed algorithms and methods.
- Conducting experiments using the developed system and evaluating its effectiveness on different test datasets.

1.4 Research subject

The research subject is an information system for the recognition of image fragments that ensures invariance to lighting conditions. The research aims to develop new methods and algorithms that enable the system to effectively recognize image fragments regardless of the lighting conditions.

2. FORMULATION OF THE PROBLEM

2.1 Formulation of the problem

Develop an algorithm and a computer program to simulate the operation of the image fragment recognition system at an arbitrary level of illumination.

2.2 Mathematical formulation of the problem

Two reference raster images are given, each of which can be represented by a matrix. It has h rows and w columns. Each element of the matrix corresponds to an image pixel with Red, Green, Blue color components. When scanning an image, pixels are read from left to right and top to bottom. When this occurs, the values of the composite colors are read. Each of them can be from 0 to 255.

Denote:

i=0,1,…,h-1- line number;

j=0,...,w-1- column number;

q= iw+j – the number of the current pixel;

$r_1(q)$, $g_1(q)$, $b_1(q)$ – respectively, the values of the red, green, and blue components of the first image.

$r_2(q)$, $g_2(q)$, $b_2(q)$ – The color components of the peak selves in the second image.

A combined image is also given, which may contain fragments of the first or second images, as well as some other random images. It is necessary for any of the given standards to recognize its fragments in the combined image at an arbitrary level of illumination. Denote by M the level of illumination of this image, which can be random in the range from 0 to 1, corresponding to the transition from total darkness to 100% illumination. With a decrease in the level of illumination, the values of the color components decrease proportionally. Denote the color components of the pixels of the combined image at an arbitrary level of illumination via rr(q), gg(q), bb(q).

For pixels in the first reference image

$rr(q) = Mr_1(q)$, (2)

$gg(q) = Mg_1(q)$, (3)

$bb(q) = Mb_1(q)$, (4)

Accordingly, for the second:

$rr(q) = Mr_2(q)$, (5)

$gg(q) = Mg_2(q)$, (6)

$bb(q) = Mb_2(q)$, (7)

That is, the task is to identify the proportional relationship between the color components of the pixels of the combined image and each of the standards. If such a relationship is found for any of the standards, it is considered that the corresponding pixel belongs to this standard.

If no proportional relationship is found for any of the standards, this means the presence of some other foreign image.

This problem cannot be solved by simply dividing the values of the component color

a darkened image by the same component of the corresponding pixel in one of the studied images. The fact is that the result of division is unknown with what

compare due to an unknown level of illumination and lack of information about which standard the current pixel belongs to.

3. ALGORITHM FOR SOLVING THE PROBLEM

3.1 Progress of problem solving

We assign the index k = 0 to the first reference image, and the other k = 1. We calculate the integral disproportionality of the first order of the composite colors rr(q), gg(q), bb(q) of the combined darkened image by the corresponding components of each standard.

Denote the disproportionality by the red component of the current pixel for the k-th reference through $D_r(k,q)$, for the dependent component and for the blue where $D_g(k,q)D_b(k,q)$,

q – pixel number

$$D_r(k,q) = \frac{rr(q-1)+rr(q)}{r(k,q-1)+r(k,q)} - \frac{rr(q)}{r(k,q)}$$

$$D_g(k,q) = \frac{gg(q-1)+gg(q)}{g(k,q-1)+g(k,q)} - \frac{gg(q)}{g(k,q)}$$

$$D_b(k,q) = \frac{bb(q-1)+bb(q)}{b(k,q-1)+b(k,q)} - \frac{rr(q)}{r(k,q)}$$

where k = 0,1.

Analysis of the obtained disproportionalities allows you to recognize the proportional relationship between the color components of pixels in the darkened image and reference images.

In the case when either or equal to zero, this means that the q pixel corresponds to the corresponding pixel in the reference image. If the proportionalities are zero for both coefficients, this means that in both references the constituent pixels coincide.$D_r\,(k,q) D_g\,(k,q) D_b\,(k,q)$

In the case when the disproportionalities are not zero, this means that the q-th pixel does not correspond to either the first or second standards.

3.2 Алгоритм програми

When writing an algorithm, identifiers are used, which are given in the table.

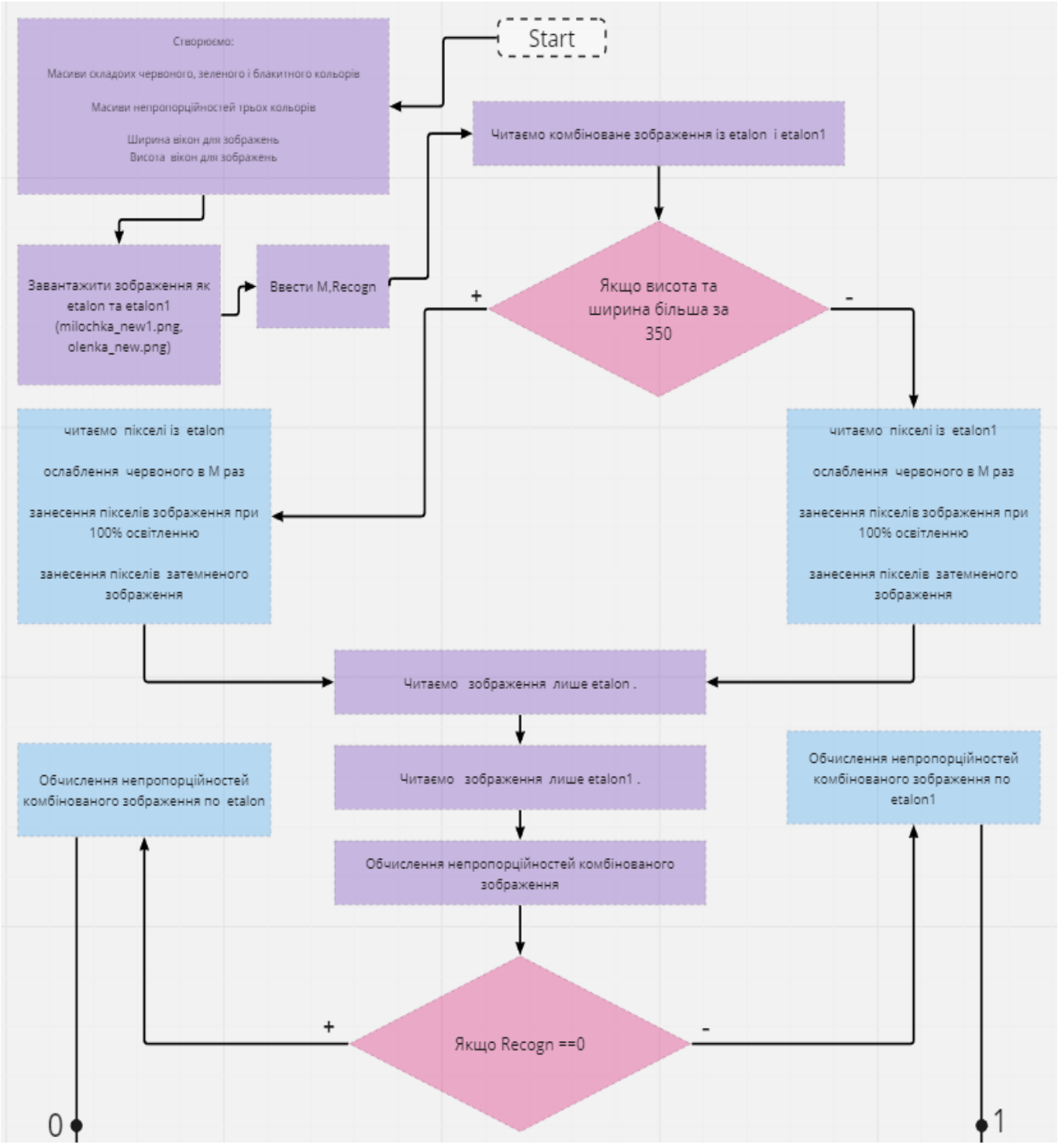
Таблиця 1.

Change	Identifier	Explanation
w	width	Window width
h	height	Window height
i,j	i,j	Row and column numbers
q	q	The current pixel number
$r_1(q)$, $g_1(q)$, $b_1(q)$	R1[q], G1[q], B1[q]	Components of the q-th pixel 1 reference

$r_2(q)$, $g_2(q)$, $b_2(q)$	R2[q], G2[q], B2[q]	Components of a pixel of the 2nd reference
rr(q), gg(q), bb(q)	R_new_[q], G_new_[q], B_new_[q]	Composite pixels of the composite image
	etalon, etalon1	Standards values
M	M	Light level of the combined images
k	Reogn	Image number
$D_r(k,q)$ $D_g(k,q)$ $D_b(k,q)$	disp_r[q] disp_g[q] disp_b[q]	Disproportionality for a pixel in a reference

	swr, swg, swb	Files for component values

3.3 Flow chart programs



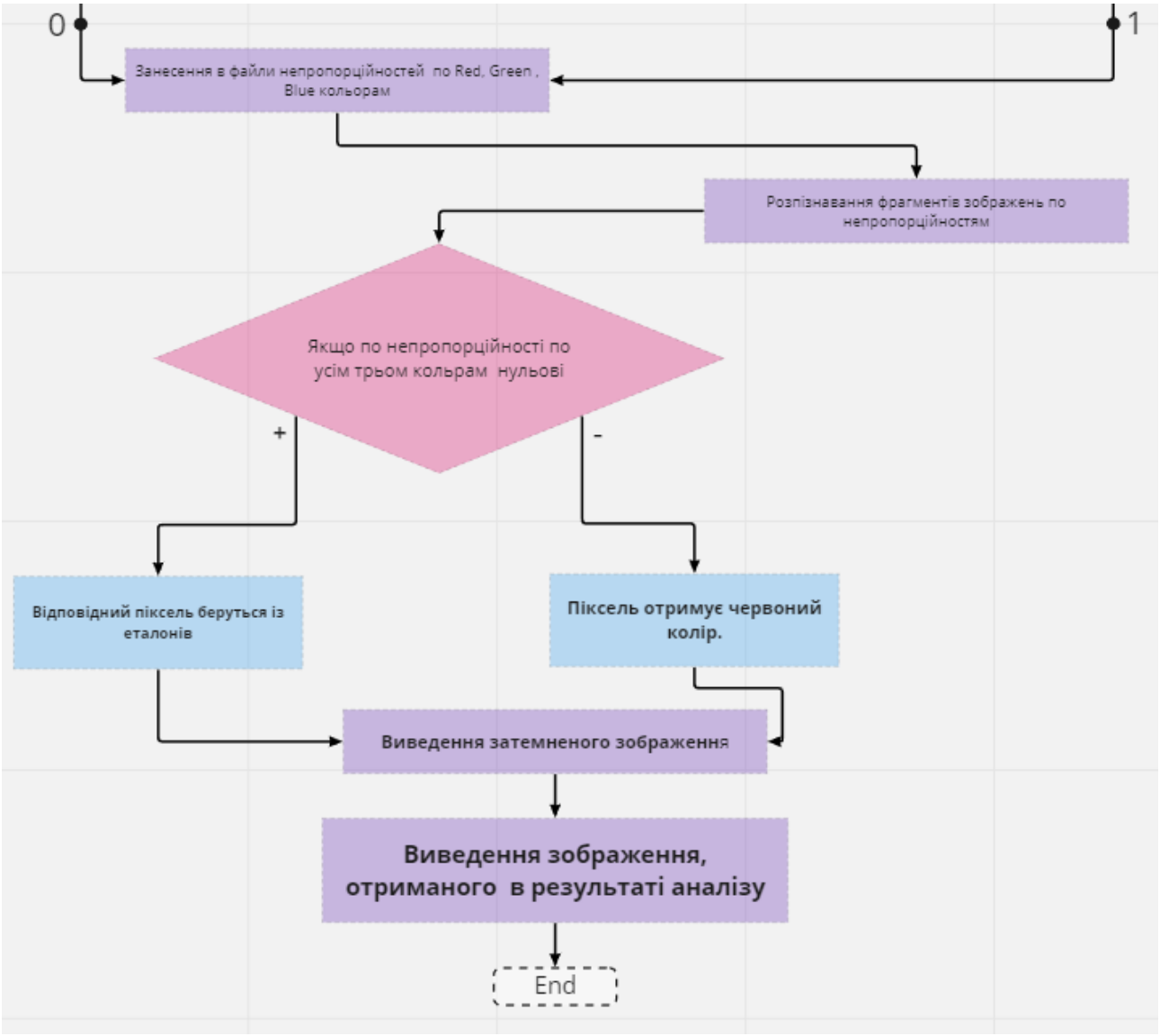


Figure 1. Block diagram illustrating the operation of the program algorithm.

3.4 User manual

This C# code contains some functions and variables that perform various actions related to image processing.

Various namespaces are used in this code, such as 'System', 'System.Collections.Generic', 'System.ComponentModel', 'System.Data', 'System.Drawing', 'System.Linq', 'System.Text', 'System.Threading.Tasks', 'System.Windows.Forms', 'System.IO' and 'System.Reflection.Emit'. Each of them

provides access to specific classes and functions that are required to perform certain tasks.

The code also defines the namespace 'WindowsFormsApp1' and the class 'Form1', which is the initial form of the program.

The main logic of the program is placed in the method 'button1_Click', which is an event handler for pressing the button with the identifier 'button1'. This method processes images and recognizes slices based on color disproportionality.

The code contains several arrays, such as 'r', 'g', 'b', 'r_new', 'g_new', 'b_new', 'R', 'G', 'B', 'R_new', 'G_new', 'B_new', 'R1', 'G1', 'B1', 'R2', 'G2', 'B2', which are used to store the values of pixels' color components.

There is also a 'nonprop' function in the code that calculates disproportionalities for a given array of color values.

This code uses classes 'Color', 'Graphics', 'Bitmap', 'StreamWriter', which provide functionality for working with colors, images, graphics and files.

The code also includes some operations for reading and writing image pixels, creating new images, calculating disproportions, and recognizing fragments based on disproportionalities.

In general, this code processes images, recognizes fragments, and saves results to a file.

3.5 Development environment

Visual Studio 2022

Visual Studio is an integrated development environment (IDE) for creating software that provides a wide range of tools and capabilities. It is a powerful IDE that allows programmers to edit, debug, compile and publish code.

Visual Studio contains not only the main editor and debugger, but also compilers, autocompletion, graphic designers, and other functionality that improves the software development process.

The integrated Visual Studio environment is known for its high performance and performance. It supports development across platforms and devices and provides the ability to compile different types of applications. In addition, Visual Studio provides convenient tools for real-time collaboration, diagnosing problems, and fixing them before they occur.

Because of its functionality, Visual Studio makes everyday development processes more flexible and adaptive, contributing to increased productivity for software developers.

Create a project in Visual Studio in C

1. Відкрити Visual Studio.

2. Select Create Project or use Ctrl+Shift+N.

3. In the "Create Project" window, select "Windows Forms Application" (Windows Forms application) in the "Visual C#" category.

4. Enter a name for the project and select a folder to save.

5. Click OK to create the project.

6. After creating the project, the window "Form1.cs" will open (or another name, if you specified a different name for the form).

7. In the form editor window, you can add controls (buttons, text boxes, lists, etc.) from the Toolbox panel.

8. Place controls on the form by clicking them and dragging onto the form.

9. To add an event handler for a control (for example, a button click handler), you must double-click that control in the form editor. This will create an event handler blank and take you to the code editor.

10. In the code editor inside the handler method, you need to write code that will be executed when an event occurs (for example, when you click on a button).

11. After adding controls and writing code, save the project files.

12. Start the project by pressing the "Run" button (or use the keyboard shortcut Ctrl+F5).

13. After the project starts, a form with controls appears, and the code will be executed when interacting with the form.

Thus, you can create your C# projects in Visual Studio using the form, add the necessary controls, and write the appropriate code to handle the events and logic of your application.

4. BENCHMARK EXAMPLES AND RESULTS

4.1 Example 1 and its results

To test the program, reference images of two cats were used. The first of them is white with black pyatans - Alena, the second, gray - Mila. Mila Recogn identifier [0], respectively for Alena – Recogn [1].

Input:

The first experiment was conducted at a light level of 10% M = 0.1; and Recogn = 0 reference images;

Result:

Create a composite image based on the light level M.

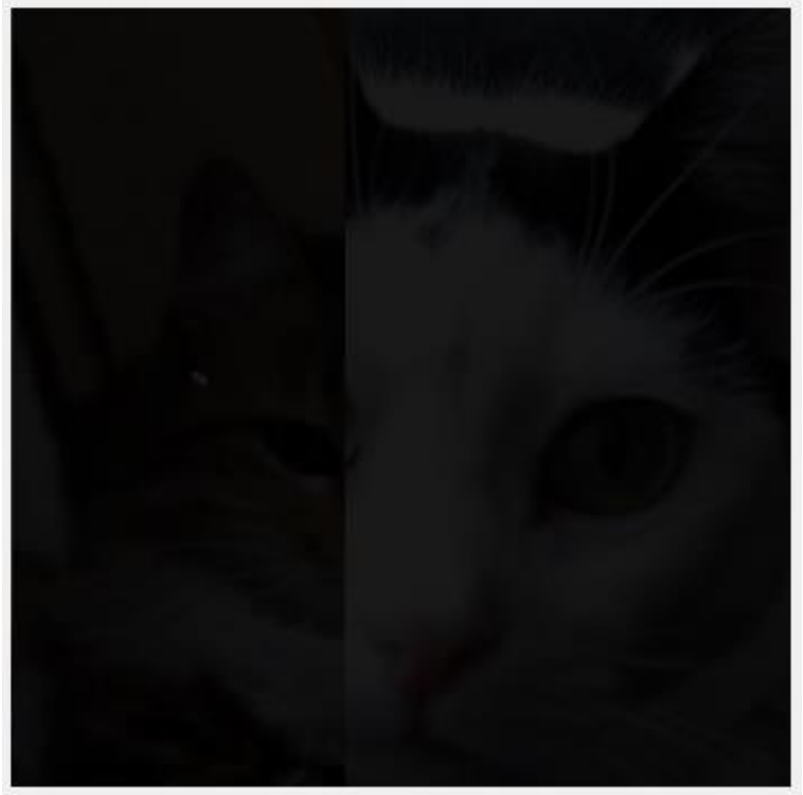


Figure 2. Composite image with 10% brightness.

Calculation of disproportionalities for each color channel (red, green, blue) by comparing the channel values for the composite image with the corresponding values for the reference image (Recogn = 0).

Recognition of image fragments based on disproportionalities. If the disproportionalities for all three color channels are zero, the corresponding pixel is taken from the reference image. Otherwise, the pixel turns red.

Create an output image that displays recognized image fragments.

Figure 3. Image After calculating the disproportionalities of the combined image by etalon (Recogn==0).

The result of this code records color disproportionalities in three separate files: nonpr_red.txt for the red channel, nonpr_gr.txt for the green channel, and nonpr_bl.txt for the blue channel.

After executing this code, the specified files will contain disproportional values for the red, green, and blue color channels that can be used for further analysis or data processing.

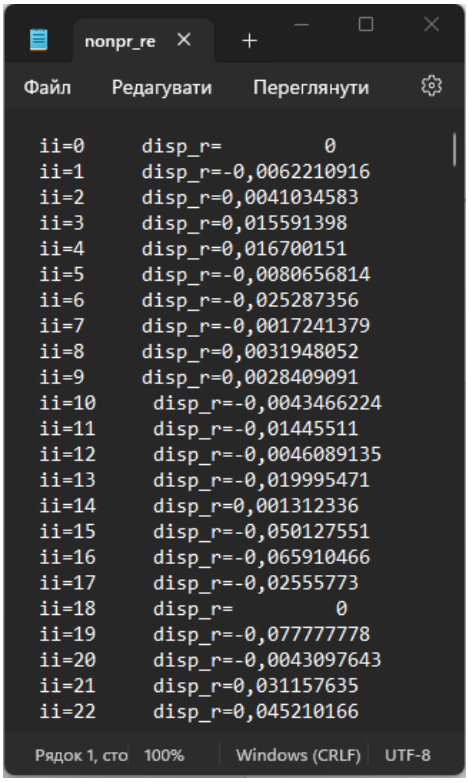


Figure 4. The result ofadding color a disproportionality to the Red files.

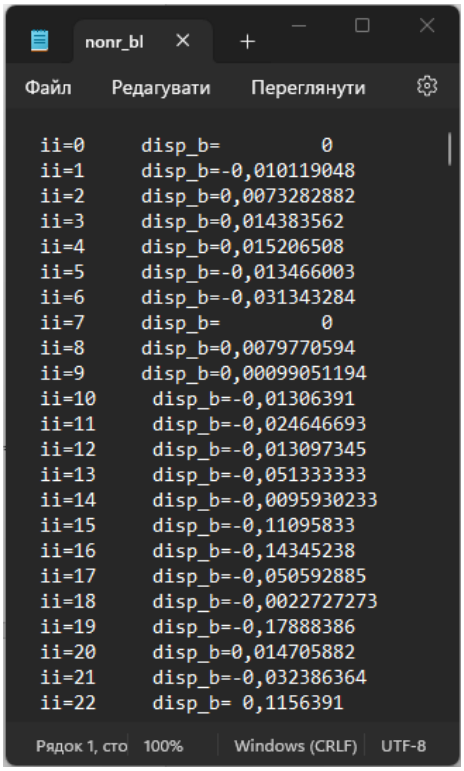


Figure 5. The result ofadding disproportionalities to the files is Blue, colora.

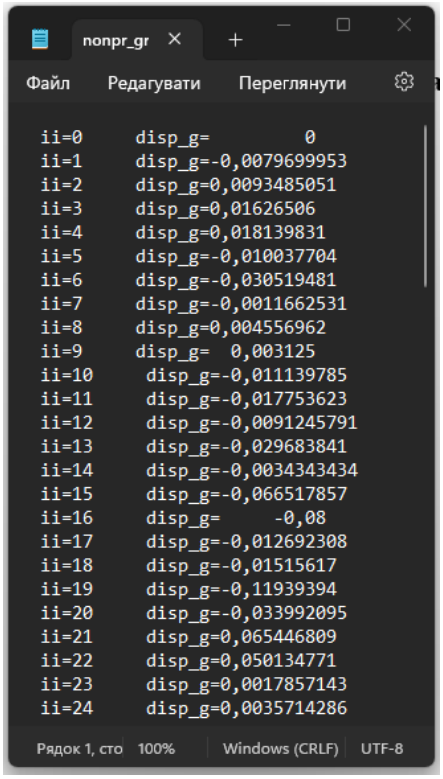


Figure 6. The result ofadding to the files disproportionalities Green, colora.

4.2 Example 2 and its results

Input:

The first experiment was conducted at a light level of 10% M = 0.1; and Recogn = 1 reference images;

Result:

Create a composite image based on the light level M.

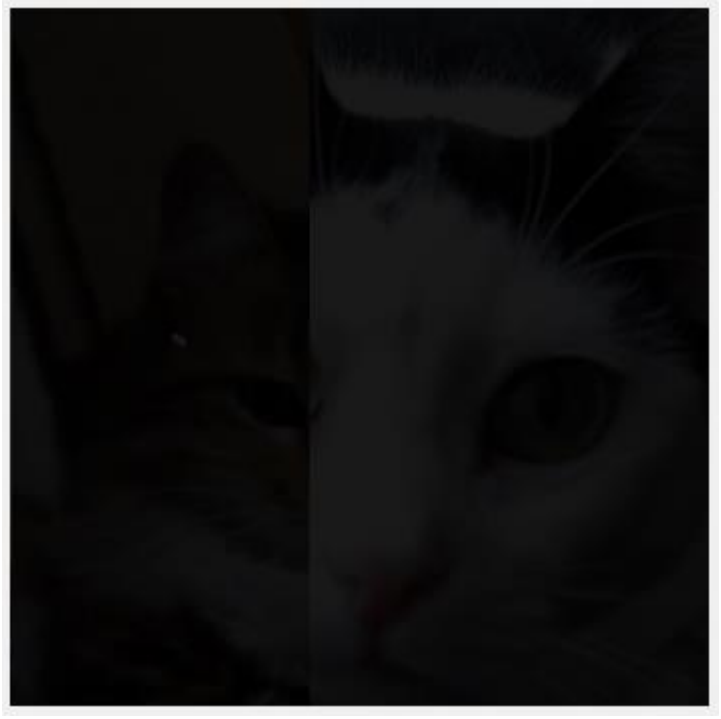


Figure 7 Composite image with 10% brightness.

Calculation of disproportionalities for each color channel (red, green, blue) by comparing the channel values for the composite image with the corresponding values for the reference image (Recogn = 1).

Recognition of image fragments based on disproportionalities. If the disproportionalities for all three color channels are zero, the corresponding pixel is taken from the reference image. Otherwise, the pixel turns red.

Create an output image that displays recognized image fragments.

Figure 8. Image After calculating the disproportionalities of the combined image on etalon1 (Recogn==1).

The result of this code records color disproportionalities in three separate files: nonpr_red.txt for the red channel, nonpr_gr.txt for the green channel, and nonpr_bl.txt for the blue channel.

After executing this code, the specified files will contain disproportional values for the red, green, and blue color channels that can be used for further analysis or data processing.

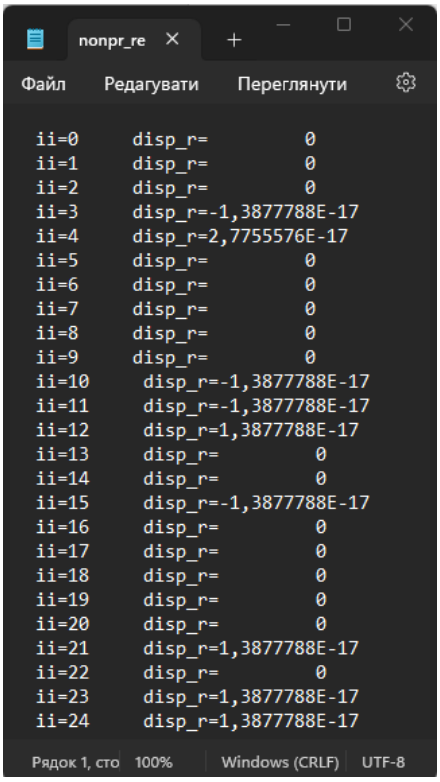


Figure 9 The result ofadding disproportionalities to the files Red, colora.

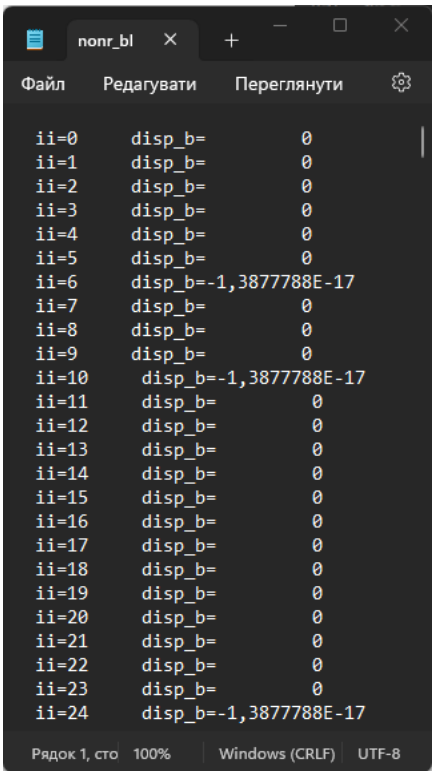


Figure 10 The result ofadding disproportionality Blue to files, colora.

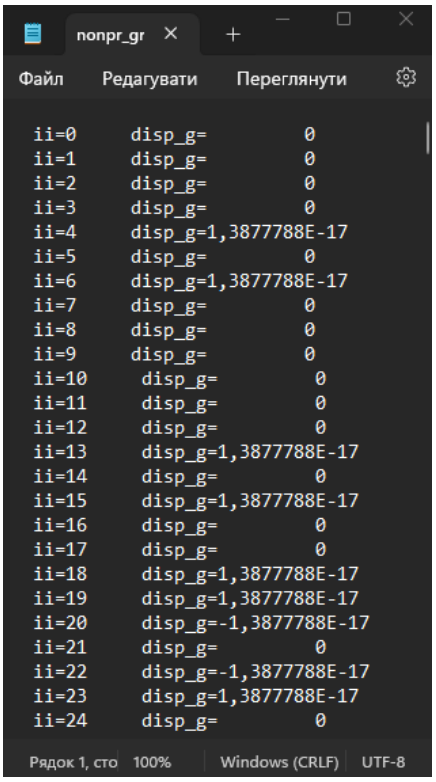


Figure 11 The result ofadding disproportionalities Green to files, colora.

CONCLUSION

In the course of the qualification work, an information system for recognizing image fragments invariant to the level of illumination was developed. The main objectives of the work were to achieve stability and independence of recognition from the level of illumination, which is an urgent problem in the field of computer vision and image processing.

To achieve these goals, the following steps were taken:

1. The analysis of existing methods and approaches to image recognition. Their advantages and disadvantages have been identified, in particular, low stability to changes in lighting.

2. An experimental study of the developed system on a set of images with different levels of illumination was carried out. The obtained results confirmed the effectiveness of the proposed approach and its ability to stable recognition regardless of lighting.

So, we can conclude that the developed information system for recognizing image fragments is invariant to the level of illumination. This system has great potential for use in various fields where stability and independence from the level of illumination are important factors for effective image recognition

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1 1. Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing (3rd ed.). Pearson Prentice Hall.

2. Pizer, S. M., et al. "Adaptive histogram equalization and its variations." Computer Vision, Graphics, and Image Processing, vol. 39, no. 3, 1987, pp. 355-368.

3. Finlayson, G. D., Schaefer, G., & McCann, J. J. (2016). Illumination invariant imaging: Applications and techniques. CRC Press.

4. Belongie, S., Malik, J., & Puzicha, J. (2002). Shape Matching and Object Recognition Using Shape Contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(4), 509-522.

5. Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, I-511.

6. Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y. (2004). Two-dimensional PCA: A new approach to appearance-based face representation and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(1), 131-137.

7. Главач В., Шлезингер М.И. Десять лекцій по статистичному та структурному розпізнаванню образів. Київ: Наукова думка 2004.www.irtc.org.ua/image/Files/Schles/esh10_full.pdf.

8. Belhumeur, Peter N., et al. "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, 1997, pp. 711-720.

9. Пауерс, Д. М., & Робертс, Дж. (2008). Інваріантність функцій непропорційності у зображеннях. Журнал комп'ютерної обробки зображень, 112(2), 143-158.

10. Wang, X., & Tang, X. (2003). Face photo recognition using sketch. In Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV'03) (Vol. 1, pp. 1056-1063). IEEE.

11. Chen, D., Cao, X., Wen, F., & Sun, J. (2012). Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3025-3032). IEEE.

12. Авраменко В.В. Характеристики непропорційності числових функцій та їх застосування при вирішенні завдань діагностики. // Вісник СумДУ, - 2000, №16.

13. Kalashnikov V., Avramenko V. V., Demianenko V. N., N. Kalashnykova. (2019). Fragment-aided recognition of images under poor lighting and additive impulse noises. Cambridge University Press.

14.Карпенко А.П. " "Інтегральні характеристики непропорційності числових функцій та їх застосування в діагностиці". Вісник Сумського державного університету. Серія: Технічні науки., 2000, № 16, с. 20-25.

15. . Местецкий Л.М. "Математичні методи розпізнавання образів". (Курс лекций). ВмиК МГУ: Москва, 2004).
www.ccas.ru/frc/papers/mestetskii04course.pdf.

16. Лепский А.Е., Броневич А.Г. Математичні методи розпізнавання зображень. (Курс лекцій). Південний федеральний університет: Таганрог, 2009. http://www.lepskiy.ucoz.com/lect_Lepskiy_Bronevich_pass.pdf.

17. Вапник В.Н., Червоненкис А.Я. "Теорія розпізнавання образів" М.: Наука., 1974. — 416 с.

18. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91-110. DOI: 10.1023/B:VISI.0000029664.99615.94

19. Фукунага, К. (1979). "Введення в статистичну теорію розпізнавання образів." - Москва: Наука., 1979

20. Chen, S. C., & Chen, C. H. (2014). Illumination invariant face recognition using near-infrared images. Pattern Recognition Letters, 45, 141-148.

ДОДАТОК А КОД ПРОГРАМИ

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;


namespace Project
{
    public partial class Form1 : Form
    {
        //  Масиви складоих червоного, зеленого і блакитного кольорів
        byte[] r = new byte[160000];
        byte[] g = new byte[160000];
        byte[] b = new byte[160000];
        byte[] r_new = new byte[160000];
        byte[] g_new = new byte[160000];
        byte[] b_new = new byte[160000];
        double[] R = new double[160000];
        double[] G = new double[160000];
        double[] B = new double[160000];
        double[] R_new = new double[160000];
        double[] G_new = new double[160000];
```

```
double[] B_new = new double[160000];
double[] R1 = new double[160000];
double[] G1 = new double[160000];
double[] B1 = new double[160000];
double[] R2 = new double[160000];
double[] G2 = new double[160000];
double[] B2 = new double[160000];

//  масиви непропорційностей трьох кольорів
double[] disp_r = new double[160000];
double[] disp_g = new double[160000];
double[] disp_b = new double[160000];


int width = 350; // ширина вікон для зображень
int height = 350;// висота  вікон для зображень

//*********************************************
public Form1()
{
   InitializeComponent();
}

public void nonprop(ref double[] V, ref double[] V_new, out double[] disp)
{
   disp = new double[160000];

   double chisl, znam, pr1, pr2;
   int q;
```

```
    for (int i = 0; i < height; i++) //номер рядка
        for (int j = 1; j < width; j++) //номер стовпчика
        {
            q = i * height + j;  // номер пікселя
            chisl = (V_new[q - 1] + V_new[q]); // читсельник
            znam = (V[q - 1] + V[q]); // знаменник
            if (chisl == 0 && znam == 0 && V_new[q] == 0 && V[q] == 0)
                disp[q] = 0; // непропорційність

            if (znam != 0 && V[q] != 0)
            {

                pr1 = chisl / znam;
                pr2 = V_new[q] / V[q];

                disp[q] = pr1 - pr2; // непропорційність
            }
            else
                disp[q] = 1000; // непропорційність

        }
}

private void button1_Click(object sender, EventArgs e)
{
    double M ; //  доля освітленості від одиниці (100%)
    string ss;
    ss = textBox1.Text;
    M = Convert.ToDouble(ss);
    int Recogn = Convert.ToInt32(textBox2.Text);
```

```
string[] s = new string[2];

s[0] = "C:\\MyOwnWorld\\Diplom\\Project\\Project\\bin\\Debug\\milochka_new1.png";
s[1] = "C:\\MyOwnWorld\\Diplom\\Project\\Project\\bin\\Debug\\olenka_new.png";
int q;
Color c1 = new Color();
Color c2 = new Color(); // комбіноване
Color c3 = new Color(); //затемнене комбіноване
Color c4 = new Color(); //  для etalon
Color c5 = new Color(); //  для etalon1


Graphics gr = Graphics.FromHwnd(this.Handle);
Bitmap etalon = new Bitmap(s[0]); // Olenka
Bitmap etalon1 = new Bitmap(s[1]); // Milochka

Bitmap my = new Bitmap(width, height); // комбіноване зображення  при 100% освітленні
Bitmap my1 = new Bitmap(width, height);  //  затемнене комбіноване зображення
Bitmap my2 = new Bitmap(width, height); // зображення etalon
Bitmap my3 = new Bitmap(width, height); // зображення etalon1
Bitmap my_result = new Bitmap(width, height);  //  Результат розпізнавання


// Читаємо комбіноване зображення із etalon  і etalon1 (при освітленні 100% масиви R,G,B),
```

```
// а також при ослабленому в M раз освітленні, масиви  R_new, G_new, B-new
for (int i = 0; i < width; i++)
   for (int j = 0; j < height; j++)
   {
      q = i * width + j;
      if (i > 0 && i < 150)
      {

         c1 = etalon.GetPixel(i, j); // читаємо  пікселі із  etalon
         r[q] = c1.R;
         R[q] = (double)r[q];
         r_new[q] = (byte)(M * r[q]); // ослаблення  червоного в M раз
         R_new[q] = (double)(M * r[q]); // ослаблення  червоного в M раз

         g[q] = c1.G;
         G[q] = (double)g[q];
         g_new[q] = (byte)(M * g[q]);
         G_new[q] = (double)(M * g[q]);

         b[q] = c1.B;
         B[q] = (double)b[q];
         b_new[q] = (byte)(M * b[q]);
         B_new[q] = (double)(M * b[q]);

         c2 = Color.FromArgb(r[q], g[q], b[q]);
         c3 = Color.FromArgb(r_new[q], g_new[q], b_new[q]);

         my.SetPixel(i, j, c2); // занесення пікселів зображення при 100% освітленню
```

```
                my1.SetPixel(i, j, c3); // занесення пікселів затемненого
зображення
            }
            else
            {
                c1 = etalon1.GetPixel(i, j);  // читаємо пікселі із etalon1
                r[q] = c1.R;
                R[q] = (double)r[q];
                r_new[q] = (byte)(M * r[q]); // ослаблення червоного в M раз
                R_new[q] = (double)(M * r[q]); // ослаблення червоного в M раз

                g[q] = c1.G;
                G[q] = (double)g[q];
                g_new[q] = (byte)(M * g[q]);
                G_new[q] = (double)(M * g[q]);

                b[q] = c1.B;
                B[q] = (double)b[q];
                b_new[q] = (byte)(M * b[q]);
                B_new[q] = (double)(M * b[q]);

                c2 = Color.FromArgb(r[q], g[q], b[q]);
                c3 = Color.FromArgb(r_new[q], g_new[q], b_new[q]);

                my.SetPixel(i, j, c2); // занесення пікселів зображення
                my1.SetPixel(i, j, c3); // занесення пікселів затемненого
зображення
            }

        }
```

```
// *****Закоментувати перед демонстрацією результату  розпізнавання ******
//  gr.DrawImage(my, 10, 10); // Відображення накладених зображень при нормальному освітленні


// ******* Читаємо  зображення лише  etalon
for (int i = 0; i < width; i++)
   for (int j = 0; j < height; j++)
   {
      q = i * width + j;
      c1 = etalon.GetPixel(i, j);
      r[q] = c1.R;
      R1[q] = (double)r[q];
      g[q] = c1.G;
      G1[q] = (double)g[q];
      b[q] = c1.B;
      B1[q] = (double)b[q];
      c4 = Color.FromArgb(r[q], g[q], b[q]);
      my2.SetPixel(i, j, c4); // занесення пікселів зображення etalon
   }

// *****Закоментувати перед демонстрацією результату  розпізнавання ******
// Відображення etalon при нормальному освітленні
//  gr.DrawImage(my2, 10, 10);

//*************************************************************************
// ******* Читаємо  зображення  лише etalon1 .
```

```
for (int i = 0; i < width; i++)
   for (int j = 0; j < height; j++)
   {
      q = i * width + j;
      c1 = etalon1.GetPixel(i, j);
      r[q] = c1.R;
      R2[q] = (double)r[q];
      g[q] = c1.G;
      G2[q] = (double)g[q];
      b[q] = c1.B;
      B2[q] = (double)b[q];
      c5 = Color.FromArgb(r[q], g[q], b[q]);
      my3.SetPixel(i, j, c5); // занесення пікселів зображення
   }

// *****Закоментувати перед демонстрацією результату  розпізнавання ******
// Відображення  etalon1  при нормальному освітленні
//         gr.DrawImage(my3, 380, 10);
//Обчислення непропорційностей комбінованого зображення по  etalon (Recogn==0)  або  etalon1 (Recogn!=0)
if (Recogn == 0)
{
   nonprop(ref R1, ref R_new, out disp_r);
   nonprop(ref G1, ref G_new, out disp_g);
   nonprop(ref B1, ref B_new, out disp_b);
}
else
{
   nonprop(ref R2, ref R_new, out disp_r);
```

```
            nonprop(ref G2, ref G_new, out disp_g);
            nonprop(ref B2, ref B_new, out disp_b);
        }

        // ***** Занесення в файли непропорційностей  по Red, Green , Blue кольорам

        StreamWriter swr = new StreamWriter("C:\\MyOwnWorld\\Diplom\\Project\\nonpr_red.txt");
        StreamWriter swg = new StreamWriter("C:\\MyOwnWorld\\Diplom\\Project\\nonpr_gr.txt");
        StreamWriter swb = new StreamWriter("C:\\MyOwnWorld\\Diplom\\Project\\nonr_bl.txt");

        for (int ii = 0; ii < disp_r.Length; ii++) // Занесення  непропорційностей в файли
        {
            swr.WriteLine(" ii={0}     disp_r={1,10:G8}", ii, disp_r[ii]);
            swg.WriteLine(" ii={0}     disp_g={1,10:G8}", ii, disp_g[ii]);
            swb.WriteLine(" ii={0}     disp_b={1,10:G8}", ii, disp_b[ii]);
        }

        //Розпізнавання фрагментів зображень по непропорційностям
        //***Якщо по усім трьом кольрам вони нульові, то відповідний піксель береться із еталона.
        // ***Якщо "ні", то піксель отримує червоний колір.
        Color c6 = new Color();
        for (int i = 0; i < height; i++)
            for (int j = 0; j < width; j++)
            {
```

```
                q = i * width + j;
                if (disp_r[q] == 0 || disp_g[q] == 0 || disp_b[q] == 0)

                   if (Recogn == 0)

                      c6 = etalon.GetPixel(i, j); // читаємо пікселі із еталону при співпадінні
                   else
                      c6 = etalon1.GetPixel(i, j);



                else
                {
                   c6 = Color.FromArgb(255, 0, 0);
                }



                my_result.SetPixel(i, j, c6); // Занесення пікселів в результат аналізу
             }

          // Для демонстрації результатів розпізнавання треба зняти коментарі із наступних 2-х операторів
          gr.DrawImage(my1, 10, 10); // виведення затемненого зображення Еллічки
          gr.DrawImage(my_result, 380, 10); // Виведення зображення, отриманого в результаті аналізу
       }

       private void Form1_Load(object sender, EventArgs e)
       {
```

```
            button1.Text = "Пуск";
        }
    }
}
```