

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Оксана ШОВКОПЛЯС

_____ (підпис)

_____ грудня 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»

освітньо-професійної програми «Інформатика»

на тему: Метод розпізнавання спотвореного 2-D зображення

здобувача групи ІН.м - 34 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 КН,
кандидат технічних наук, доцент

Віктор

АВРАМЕНКО

_____ (підпис)

Консультант з англійської мови

доцент кафедри КН,
кандидат фізико-математичних наук,
доцент

**Оксана
ШОВКОПЛЯС**

_____ (підпис)

Суми – 2024

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

«Затверджую»

В.о. завідувача кафедри

Оксана ШОВКОПЛЯС

(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття освітнього ступеня магістра

зі спеціальності 122 «Комп'ютерні науки», освітньо-професійної програми «Інформатика»
здобувача групи ІН.м-34 Винник Катерини Євгенівни

1. Тема роботи: Метод розпізнавання спотвореного 2-D зображення

затверджую наказом по СумДУ від «03» грудня 2024 року № 1257-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 06 грудня 2024 року

3. Вхідні дані до кваліфікаційної роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.

2) Попередні дослідження з теми 3) Розробка програми 4) Аналіз результатів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти до проекту (роботи), із зазначенням 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	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>	19.00 - 10.10	
3	<i>Розробка програми</i>	05.11 - 29.11	
4	<i>Аналіз отриманих результатів</i>	29.11 - 30.11	
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>	01.12 - 06.12	

Здобувач вищої освіти _____
(підпис)

Керівник _____
(підпис)

АНОТАЦІЯ

Записка: 34 стор., 4 рис., 1 додаток, 29 використаних джерел.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи є актуальною, оскільки присвячена розв'язанню важливої практичної задачі розпізнавання двовимірних зображень, спотворених внаслідок їх проходження через нелінійні об'єкти. Ця задача має важливе значення в багатьох галузях, зокрема в комп'ютерному зорі, обробці зображень, медичних дослідженнях та автоматизації процесів, що вимагають високої точності аналізу візуальної інформації.

Об'єкт дослідження — Процес розпізнавання спотворених двовимірних зображень.

Мета роботи — Розробка методу розпізнавання двовимірних зображень, які зазнали спотворення внаслідок проходження через нелінійний об'єкт, а також створення алгоритму та комп'ютерної програми для ілюстрації роботи запропонованого методу.

Методи дослідження — У процесі дослідження застосовано методи розпізнавання зображень, які базуються на функціях непропорційності. Ці методи використовуються для аналізу та ідентифікації двовимірних зображень, спотворених під впливом нелінійних об'єктів.

Результати — Розроблено інформаційну систему, яка зчитує дані зображень, що зазнали спотворення, виконує аналіз та корекцію спотворень, реалізує розпізнавання вихідного зображення

ЗМІСТ

1 ЛІТЕРАТУРНИЙ ОГЛЯД	5
1.1 Попередні дослідження по темі.	5
1.1.1 Використання нейронних мереж	5
1.1.2 Алгоритми пошуку відповідності	5
1.1.3 Реконструкція зображення	5
1.2 Основні відомості з теми	6
1.3 Функції непропорційності	7
1.4 Огляд методів розпізнавання не спотворених еталонних сигналів з наявністю імпульсних завад	10
1.5 Приклад розрахунку SDF непропорційності	12
1.6 Алгоритм розпізнавання	13
1.7 Приклад розпізнавання спотвореного 2-D зображення	14
1.8 Середовище розробки програми	17
2 ВИБІР МЕТОДУ І ХІД РОЗВ'ЯЗАННЯ ЗАДАЧІ	18
2.1 Постановка задачі	18
2.2 Математична постановка задачі	18
2.3 Вибір методу розв'язання задачі	20
2.4 Хід розв'язання задачі	21
2.5 Алгоритм розв'язання задачі	22
2.6 Інструкція для користувача	25
3. КОНТРОЛЬНИЙ ПРИКЛАД	25
ВИСНОВКИ	28
СПИСОК ЛІТЕРАТУРИ	29
Додаток	32

1 ЛІТЕРАТУРНИЙ ОГЛЯД

1.1 Попередні дослідження по темі.

Розпізнавання порушених двовимірних (2D) зображень є важливим обов'язком на території комп'ютерного зору, уточнення зображення та машинного інтелекту. Ця проблема часто виникає в кількох областях, як-от перевірка особи, робота роботів, медична діагностика та обробка даних космічного корабля [1, 2].

1.1.1 Використання нейронних мереж

Глибоке навчання привернуло значну увагу до прогресу згорткових нейронних мереж (CNN). Простий і зрозумілий метод: автоматичний вибір ознак + висока ефективність розпізнавання з нелінійними змінами [3]. Водночас складність моделювання може призвести до збільшення обчислювальної складності та потребувати значної кількості даних для навчання [4].

1.1.2 Алгоритми пошуку відповідності

Такі методи, як SIFT і SURF, які ретельно вивчають життєво важливі аспекти та дескриптори, забезпечують виняткову точність у ідентифікації місцевих елементів, незалежно від значних змін [5]. Зіставлення та реконструкція зображень можуть бути ефективно виконані за допомогою цих підходів [6].

1.1.3 Реконструкція зображення

Для завдань, що вимагають відновлення спотворених зображень, застосовуються методи регресії та інтерполяції [7]. Використання як стандартних алгоритмів, так і методів машинного навчання значно підвищує якість реконструкції [8].

1.2 Основні відомості з теми

Існує широкий спектр задач, для вирішення яких необхідне розпізнавання стандартних сигналів. Крім того, зростає потреба у розпізнаванні фрагментів стандартних зображень, які надходять у реальному часі від камер відеоспостереження [9].

На практиці такі задачі значно ускладнюються низкою факторів, зокрема необхідністю оперативного аналізу даних, фрагментарністю вхідного сигналу, варіативністю амплітуд сигналів залежно від умов прийому-передачі та нелінійними спотвореннями, що виникають у пристроях зі статичними характеристиками, які мають нелінійні ділянки (наприклад, відео підсилювачах) [10]. Невідомі параметри нелінійності або випадкові зміни цих параметрів у часі також ускладнюють задачу. Усе це обумовлює необхідність створення методів, інваріантних до амплітуди стандартного сигналу, здатних розпізнавати фрагменти зображень навіть за наявності випадкових завад і нелінійних спотворень [9].

Серед підходів до розпізнавання зображень використовуються нормалізація для афінних спотворень [11], ймовірнісні моделі [12], обробка вагових значень у бездротових мережах [13] та методи каскадних класифікаторів [14]. Широко застосовуються алгоритми на основі нейронних мереж [15–17] і вейвлет-аналіз [18]. Проте всі ці методи мають свої обмеження, такі як необхідність великих обчислювальних ресурсів або об'ємних баз даних.

Ефективність розпізнавання фрагментів стандартних сигналів за наявності імпульсних шумів розглядалася в [19], але без врахування нелінійних спотворень. У роботах [20, 21] запропоновано принцип розпізнавання спотворених зображень з використанням функцій непропорційності, які дозволяють визначати моменти зникнення завад та розпізнавати стандартні сигнали навіть за умов випадкових спотворень [9]. У разі двовимірних зображень вважається, що вони мають однаковий розмір і не піддаються обертанням, що відповідає умовам використання стаціонарних камер при слабкому освітленні.

1.3 Функції непропорційності

Функції непропорційності є характеристиками числових функцій. Вони дозволяють кількісно визначити відхилення співвідношення між двома числовими функціями від пропорційного співвідношення [9].

Розрізняють наступні типи непропорційностей [20]:

- непропорційність по похідній n - порядку;
- непропорційність по значенню n - порядку;
- відносна непропорційність n - го порядку;
- інтегральна непропорційність n -го порядку [17].

Нижче наведено короткий виклад функцій непропорційності, які використовуються в звіті.

Якщо існує пропорційна залежність, усі ці непропорційності дорівнюють нулю, незалежно від значення коефіцієнта пропорційності.

Отже, функція непропорційності по похідній n - порядку має вигляд:

$$@d_x^{(n)}y = \frac{y}{x^n} - \frac{1}{n!} \cdot \frac{d^n y}{dx^n} \quad (1.1)$$

Для степеневі функції ця непропорційність дорівнює нулю $y=kx^n$ незалежно від коефіцієнту k . тут $n \geq 1$ є цілим числом.

для $n = 1$:

$$@d_x^{(1)}y = \frac{y}{x} - \frac{dy}{dx} \quad (1.2)$$

Символ @ обрано для позначення операції обчислення непропорційності, d - від похідного англ. "derivative". Ліва частина виразу 2 читається як "at d one y by x".

Для функцій $\psi(t) = k\varphi(t)$, які визначаються параметрично (t є параметром), непропорційність по похідній 1-го порядку (1.2) має вигляд:

$$@d_x^{(1)}y = @d_{\varphi(1)}^{(1)}\psi(t) = \frac{y}{x} - \frac{y'}{x'_t} = \frac{\psi(t)}{\varphi(1)} - \frac{\psi'(t)}{\varphi'(1)} \quad (1.3)$$

Очевидно, якщо $\psi(t) = k\phi(t)$, то непропорційність (1.3) дорівнює нулю на всій області існування функції $\phi(t)$, незалежно від значення k .

Легко переконатися, що функції непропорційності мають такі властивості:

1. Множення функції на будь-який скаляр призводить до множення його функції непропорційності на той самий скаляр.

2. Функція непропорційності суми (різниці) числових функцій дорівнює сумі (різниці) їх функцій непропорційності.

Зауваження 1. Іншими словами, оператор $@d_x^{(n)}$ визначений на просторі $C^n(\Omega)$ n раз безперервно диференційованих дійсних функцій є лінійним на цьому просторі.

Для виявлення поліноміальних залежностей використовується так звана послідовна пропорційна функція за похідною першого порядку [15].

Це послідовний розрахунок непропорційності для раніше розрахованої непропорційності. Наприклад, непропорційність (1.3) спочатку обчислили відносно похідної першого порядку функції $i(x)$ щодо x .

Відбувається послідовне обчислення (C - разів) непропорційностей. В результаті отримується C - непропорційність (SDF) відносно похідної 1-го порядку.

Наприклад, для $C = 3$ відбувається послідовне обчислення згідно (1.4):

$$@d_x^{(3)}y(t) := @d_x^{(1)} \left\{ d_x^{(1)} \left[d_x^{(1)} y(t) \right] \right\} \quad (1.4)$$

Обчислення послідовності непропорційності дозволяє визначити невідомі коефіцієнти в поліномі. Їхнє обчислення виконується за кроками, кількість яких дорівнює степеню полінома [9]. Це можна побачити на прикладі розрахунку C -непропорційності (1.4) для полінома третього порядку $y = a_3x^3 + a_2x^2 + a_1x$.

$$\text{Крок 1: } c_1 = @d_x^{(1)}y = \frac{y}{x} - y' = -2a_3x^2 - a_2x$$

$$\text{Крок 2: } c_2 = @d_x^{(1)} c_1 = \frac{c_1}{x} - c_1' = 2a_3x$$

$$\text{Крок 3: } c_3 = @d_x^{(1)} c_2 = \frac{c_2}{x} - c_2' = 0$$

На останньому кроці непропорційність дорівнює нулю, незалежно від значень коефіцієнтів полінома. Це може бути використано при розпізнаванні спотворених сигналів

При розпізнаванні сигналу використовується рівність SDF нулю.

1.4 Огляд методів розпізнавання не спотворених еталонних сигналів з наявністю імпульсних завад

В [19] приведено алгоритм розпізнавання не спотворених еталонних сигналів при наявності імпульсних завад.

Відома кінечна множина еталонних сигналів, що описуються функціями $f_i(t)$, де $t \in [0; T_i]$, $i = 1, 2, \dots, M$.

Ці функції є гладкими та неперервними, а також мають перші похідні. У випадку, коли присутня адитивна завада, сигнал, що підлягає аналізу, можна представити у вигляді наступного рівняння:

$$y(t) = kf(t + \tau_i) + \eta(t) \quad (1.5)$$

де $f_i(t)$ є i -я стандартна функція;

$\tau_i \in [0; T_i]$ - часовий зсув між сигналом та i -м еталоном;

$\eta^{(t)}$ - адитивна завада, про яку відомо лише те, що вона може зникати і з'являтися у випадкові моменти часу;

k - це коефіцієнт, значення якого невідоме.

Для мультиплікативної завади:

$$y(t) = kf_i(t + \tau_i)\eta(t) \quad (1.6)$$

По поточному значенню $y(t)$ та його першій похідній в [20] розпізнається, який конкретно еталонний сигнал передається в поточний момент. У випадках, коли сигнал описується рівняннями (1.5) або (1.6), для цього застосовувалася функція непропорційності по похідній першого порядку для числових функцій, які задані параметрично [21].

В [19] непропорційність функції (1.5) відносно еталонної функції $f_j(t + \tau_j)$ описується виразом:

$$@d_{f(t+\tau_i)}y(t) = \frac{kf_i(t+\tau_i)+\eta(t)}{f_j(t+\tau_j)} - \frac{kf_i(t+\tau_i)+\eta(t)}{f_j(t+\tau_j)} = @d_{f(t+\tau_i)}f_j(t + \tau_j) \quad (1.7)$$

На той випадок, коли $j = i$, непропорційність (1.7) має вигляд:

$$@d_{f(t+\tau_i)}y(t) = \frac{\eta(t)}{f_i(t + \tau_i)} - \frac{\eta'(t)}{f_i'(t + \tau_i)} = @d_{f_i(t+\tau_i)}\eta(t) \quad (1.8)$$

Очевидно, коли завада зникає, $\eta(t) = 0$, $\eta'(t) = 0$, із правильно вибраним зсувом $\tau_i \in [0; T_i]$, (1.8) рівняється нулю. Таким чином, те, що рівняння (1.7) дорівнює нулю, свідчить про те, що в момент часу t завада зникла, і еталонна функція $f_i(t)$ зміщена τ_i на присутня в аналізованому сигналі. Для інших еталонних функцій непропорційності (1.7) не будуть дорівнювати нулю при будь-якому зсуву в часі.

Винятком може бути випадок, коли декілька еталонів мають співпадаючі фрагменти.

За наявності мультиплікативної завади, аналізований сигнал описується виразом (1.6), непропорційності (1.3) $i(t)$ щодо $f_i(t + \tau)$ для зсуву часу τ_i що надає йому вигляд [19]:

$$@d^{(1)}f(t + \tau_i) y(t) = -k\eta'(t) \frac{f_i(t + \tau_i)}{f_i'(t + \tau_i)} \quad (1.9)$$

У момент, коли похідна завади $\eta'(t) = 0$, непропорційність (1.9) стає нульовою. Тобто в цьому випадку розпізнавання стандартного сигналу відбувається за наявності перешкод.

1.5 Приклад розрахунку SDF непропорційності

Розглянемо поліном (1.10), коли $p = 3$.

$$y(t) = a_3f_i^3(t) + a_2f_i^2(t) + a_1f_i(t) \quad (1.10)$$

Крок 1. Розраховується непропорційність $z_1(t)$ (1.4) функції $y(t)$ відносно $f_i(t)$:

$$z_1(t) = @d_{f_i(t)}^{(1)}y(t) = \frac{y(t)}{f_i(t)} - \frac{y'(t)}{f_i'(t)} = -2a_3f_i^2(t) - a_2f_i(t) \quad (1.11)$$

Крок 2. Розраховується непропорційність $z_2(t)$ (1.4) функції $z_1(t)$ відносно $f_i(t)$:

$$z_2(t) = @d_{f_i(t)}^{(1)} z_1(t) = \frac{z_1(t)}{f_i(t)} - \frac{z_1'(t)}{f_i'(t)} = 2a_3 f_i(t) \quad (1.12)$$

Крок 3. Розраховується непропорційність $z_3(t)$ (1.4) функції $z_2(t)$ по відношенню до $f_i(t)$:

$$SDF(t) = z_3(t) = @d_{f_i(t)}^{(1)} z_2(t) = \frac{z_2(t)}{f_i(t)} - \frac{z_2'(t)}{f_i'(t)} = 0 \quad (1.13)$$

Рівність нулю $SDF(t)$ (1.13) свідчить про те що конкретно $f_i(t)$ спотворюється нелінійним об'єктом.

1.6 Алгоритм розпізнавання

Як приклад розглянемо розпізнавання фрагментів еталонних зображень за червоною складовою кольору пікселів аналізованого зображення.

Крок 1. Встановити близьке до нуля число ε . Його модуль буде використовуватися для порівняння значення функції непропорційності при розпізнаванні.

Крок 2. Зчитати значення червоної яскравості пікселів, що належать до всіх еталонних зображень.

Крок 3. Вибирати еталонне зображення і прочитати червоні компоненти його пікселів $r_k(q)$, $1 \leq q \leq N$.

Крок 4. Обчислити функцію непропорційності $SDF(q)$ з r_k по відношенню до R_k за (1.5) і зберегти його значення як масив:

$$D_k^P(q) = SDF(q), q = 2, K, N$$

Крок 5. Розпізнавання якщо $|D_k^P(q)| \leq \varepsilon$, це означає, що q піксель аналізованого зображення можна ідентифікувати як q -ий піксель еталонного зображення. З метою зберегти його можна скопіювати в q піксель отриманого зображення.

Інакше, якщо $|D_k^P(q)| > \varepsilon$, відповідний піксель результату зображення залишається порожнім. Ця процедура повторюється для кожного пікселя.

1.7 Приклад розпізнавання спотвореного 2-D зображення

В [9] імітується система технічного зору, в якій без попередньої обробки використовується компоненти кольору: Червоний, Зелений і Синій. Також вважається, що масштаби еталонного та аналізованого зображень однакові, а обертання відсутнє.

Крім того через погане освітлення об'єкта більша частина сигналу потрапляє в діапазон яскравості від 1 до 15 (максимум 255), який називається «Чорніше чорного» (Blacker Than Black, BTV) [9]. Цей діапазон часто не використовується, і, наприклад, для відеопідсилювачів імпульси синхронізації розташовуються у відповідному діапазоні напруг. При необхідності можна використати і цей діапазон яскравості якщо використати представлений у даній роботі алгоритм [27].

Сигнал від світлодіодного матричного фотоприймача надходить на вхід транзистора, і біля нульового значення вхідного сигналу на вольт-амперній характеристиці (ВА) р-п-переходу виникає нелінійна область [28].

Вважається, що вхідний сигнал не виходить за межі нелінійної ділянки вольт-амперної характеристики транзистора.

При зниженні рівня освітленості об'єкта зберігається пропорційне співвідношення між яскравістю зображення, що надходить на світлодіодну матрицю, і його еталоном [29]. Однак після проходження через нелінійність сигнал спотворюється. Мета моделювання – показати принципову можливість розпізнавання фрагмента еталонного зображення з спотвореного та ослабленого сигналу.

Частина ВТВ становить близько 6% від усієї шкали яскравості. Інтервал не лінійної ділянки вольт-амперної характеристики приймемо 5%.

У нормованому вигляді сигнал на вході транзистора може змінюватися від 0 до 0,06 в межах нелінійної ВАХ, тобто від 0 до 0,05. Розглянемо випадок, коли нелінійна частина вольт-амперної характеристики має вигляд рис. 1.

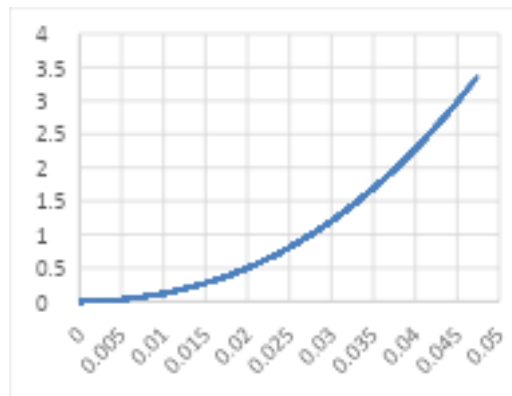


Рис 1. Нелінійна частина вольт-амперної характеристики

Він описується поліномом 3-го порядку:

$$y = x + 1000x^2 + 10000x^3,$$

де x - відносне значення вхідного сигналу (червона складова еталонного зображення);

y – параметр на виході нелінійного пристрою.

Потім цей параметр зменшується до абсолютних значень яскравості Red_new , щоб сигнал, перетворений після того, як нелінійний пристрій потрапив у більш чорну область, шляхом множення, у цьому прикладі, на 255/100. Для них розраховується непропорційність SDF (15) за наведеним вище алгоритмом [9]. Під час його обчислення були виключені пікселі, для яких Red_new дорівнює нулю або для яких відбулося ділення на нуль. Допустима розбіжність $\varepsilon = 10^{-4}$.

На рис. 2 показано результати розпізнавання для $\epsilon = 10^{-4}$ (верх) і для $\epsilon = 10^{-8}$ (внизу).

У нижній частині кожного зображення показано спотворений сигнал, отриманий для розпізнавання. Розпізнане еталонне зображення показано у верхній частині кожного зображення. Кількість пікселів, використаних і виключених з аналізу, також показано на кожному зображенні.

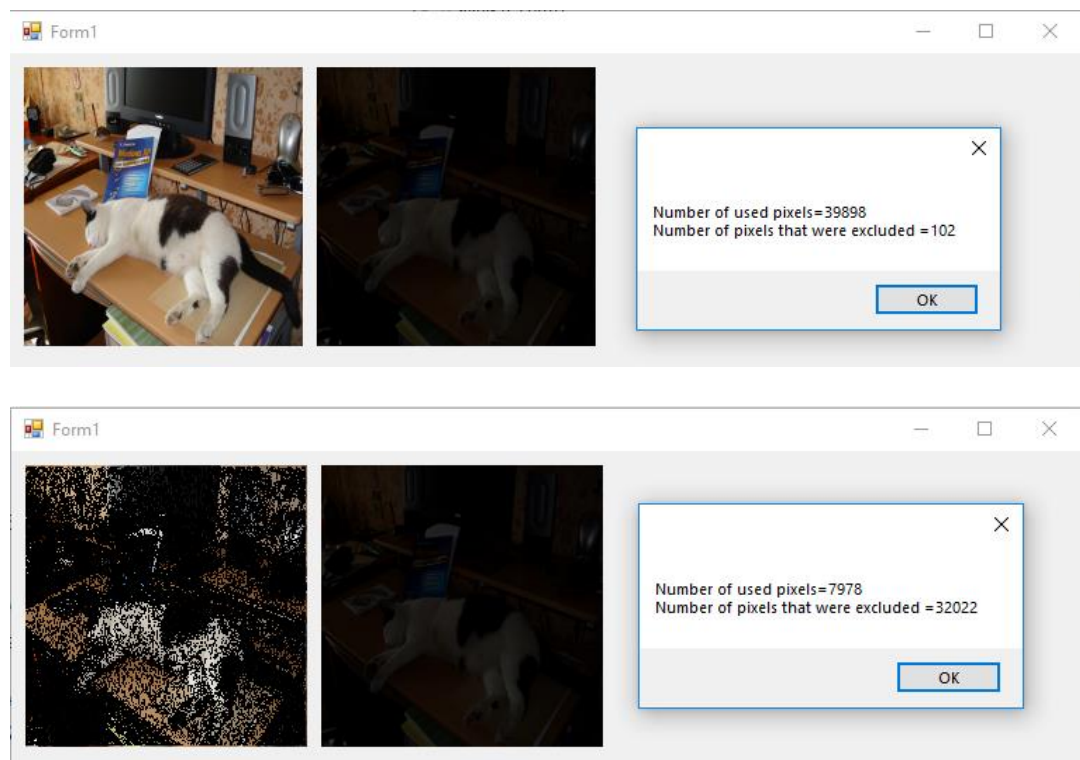


Рис 2. Спотворені та розпізнані зображення для $\epsilon = 10^{-4}$ (верх) і для $\epsilon = 10^{-8}$ (внизу).

Порівняння результатів демонструє, що зменшення значення ϵ призвело до формування значно темнішого розпізнаного зображення. Це підкреслює важливість коректного вибору порогового значення ϵ для забезпечення оптимальної якості відновлення зображення.

1.8 Середовище розробки програми

Найкраща програма для реалізації Microsoft Visual Studio. Данна програма має прості у використанні інструменти розробки графічного інтерфейсу

C# та бібліотека .NET Framework мають потужні засоби обробки зображень, такі як клас Bitmap, що використовується у додатку для завантаження, зміни пікселів та відображення зображень.

Є ще один важливий фактор - Visual Studio має найбільшу та найповнішу базу документації, а також велику спільноту розробників, що дає можливість дуже швидко отримати необхідні знання.

Все це робить Visual Studio C# найкращим фреймворком для програми обробки зображень.

2 ВИБІР МЕТОДУ І ХІД РОЗВ'ЯЗАННЯ ЗАДАЧІ

2.1 Постановка задачі

Розробити метод розпізнавання 2-D зображення, спотвореного внаслідок його обробки нелінійним об'єктом. Створити алгоритм і комп'ютерну програму для ілюстрації роботи запропонованого методу.

2.2 Математична постановка задачі

У випадку застосування відеокамери, зображення представлені у вигляді двовимірних масивів пікселів. Кожен піксель має своє значення кольору, яке визначається інтенсивністю червоної, зеленої та синьої складових. Значення кожної з цих складових може варіюватися від 0 до 255.

Дано m еталонних зображень, представлених матрицями пікселів. Після сканування зображень отримуються масиви $R_k = \{R_k(q) | 1 \leq q \leq N\}$, $G_k = \{G_k(q) | 1 \leq q \leq N\}$, $B_k = \{B_k(q) | 1 \leq q \leq N\}$, які відповідно описують значення складових яскравостей червоного, зеленого та блакитного кольорів для кожного пікселя зображення та для кожного еталону[9].

Позначення приведені в таблиці нижче:

$k = 1, 2, \dots, m$	- порядковий номер стандарту;
$q = i * w + j$	- порядковий номер пікселя на екрані;
w	- це кількість пікселів в одному рядку $w \geq 1$;
j	- це порядковий номер пікселя в рядку $1 \leq j \leq w$;
h	- загальна кількість рядків на екрані;
I	- це номер рядка $0 \leq i \leq h$;
$N = (1 + h) * w$	- це загальна кількість пікселів на екрані.

Аналогічно, отримані червоні, зелені та сині компоненти пікселів аналізованого зображення впорядковуються в окремі масиви:

$$r_k = \{r_k(q) | 1 \leq q \leq N\},$$

$$g_k = \{g_k(q) | 1 \leq q \leq N\}$$

$$b_k = \{b_k(q) | 1 \leq q \leq N\} \text{ відповідно } (k = 1, 2, \dots, m).$$

Припустимо, що нелінійна частина статичної характеристики нелінійного об'єкта описується поліномом степеня (n) з нульовим вільним членом. Якщо піксель на аналізованому зображенні відповідає пікселю з (k)-го еталонного зображення, їх значення яскравості червоного, зеленого та синього кольорів, враховуючи нелінійні спотворення, описуються поліномами з невідомими коефіцієнтами [9]. В подальшому з метою спрощення будемо розглядати лише компонент r_k . Наприклад, для ($m = 3$) ці поліноми можуть мати вигляд:

$$r_k(q) = a_{k3}^r R_k^3(q) + a_{k2}^r R_k^2(q) + a_{k1}^r R_k(q), 1 \leq q \leq N \quad (2.1)$$

де $a_{k3}^r, a_{k2}^r, a_{k1}^r, k = 1, K, m$ - невідомі коефіцієнти.

Аналогічно для складових g, b невідомі коефіцієнти $a_{k3}^g, a_{k2}^g, a_{k1}^g, a_{k3}^b, a_{k2}^b, a_{k1}^b$.

Таким чином, завдання полягає в розпізнаванні фрагмента еталонного зображення на основі поточних значень яскравості його пікселів і пікселів еталонних зображень. Це завдання потрібно виконати, незважаючи на невідомі параметри нелінійної частини статичної характеристики пристрою, який спотворює зображення.

2.3 Вибір методу розв'язання задачі

Літературний огляд свідчить що для розв'язання поставленої задачі можна застосувати алгоритм, який базується на використанні функції непропорційності [9, 8]. При послідовному обчисленню непропорційностей на n -му кроці SDF (1.10) буде дорівнювати нулю.

Однак слід зазначити, що яскравість пікселів еталонних сигналів вимірюється в цілих числах від 0 до 255. Тому замість непропорційності третього порядку, яка використовується в розрахунку SDF для неперервних функцій, можна використовувати інтегральну непропорційність першого порядку [17].

2.4 Хід розв'язання задачі

У зв'язку з тим що нелінійний об'єкт описується поліномом третього порядку розв'язання задачі відбувається в три кроки.

Крок 1. Обчислюємо інтегральну непропорційність червоної складової $r_k(q)$ яка надходить до системи розпізнавання, по відношенню до червоного компонента k -го еталонного зображення $R_k(q)$:

$$I_1(q) = @I_{P_k}^{(1)} r_k(q) = \frac{r_k(q-1) + r_k(q)}{R_k(q-1) + R_k(q)} - \frac{r_k(q)}{R_k(q)} \quad (2.2)$$

Крок 2. Обчислюється інтегральна непропорційність $I_1(q)$ по відношенню до $R_k(q)$:

$$I_2(q) = @I_{R_k}^{(1)} i_1(q) = \frac{i_1(q-1) + i_1(q)}{R_k(q-1) + R_k(q)} - \frac{i_1(q)}{R_k(q)} \quad (2.3)$$

Крок 3. Обчислюється інтегральна непропорційність $I_2(q)$ по відношенню $R_k(q)$:

$$SDF(q) = I_3(q) = @I_{R_k}^{(1)} I_2(q) = \frac{I_2(q-1) + I_2(q)}{R_k(q-1) + R_k(q)} - \frac{I_2(q)}{R_k(q)} \quad (2.4)$$

Якщо на останньому кроці непропорційність (2.4) дорівнює нулю, це означає, що відповідний піксель (k)-го еталонного зображення відповідає (q)-му пікселю аналізованого зображення.

Однак на практиці під час обчислення послідовної непропорційності $SDF(q)$ можуть накопичуватися помилки округлення. У результаті на останньому етапі можна отримати значення, яке близьке до нуля, але не є точно нульовим. Тому результат обчислення на останньому етапі необхідно порівняти з деяким числом ϵ , яке близьке до нуля. Якщо модуль цієї непропорційності менший або дорівнює ϵ , вважається, що розпізнавання відбулося успішно.

Якщо $SDF(q)$ (2.4) на третьому кроці не дорівнює нулю, треба змінити номер еталонного зображення $k = k + 1$.

При $k \leq m$, перейти до нового еталонного зображення. Інакше переходимо до іншого пікселя. У разі закінчення пікселів зображення яке аналізується, завершуємо розпізнавання і показуємо результат.

2.5 Алгоритм розв'язання задачі

При написанні алгоритму застосовуються ідентифікатори, які приведені в таблиці

Таблиця 1.

Індикатор	Тип	Призначення
my	Bitmap	Зображення, в якому зберігаються оброблені пікселі (розміру 200x200).
my_result	Bitmap	Зображення, яке містить результати обробки (також 200x200).
Red	double[]	Масив, що зберігає значення червоного кольору для кожного пікселя (розмір 160000).
Red_new	double[]	Масив, що зберігає нові значення червоного кольору після обробки (розмір 160000).

nonprop	double[]	Масив, що зберігає значення показника nonprop для кожного пікселя (розмір 160000).
nonprop2	double[]	Масив, що зберігає значення показника nonprop2 для кожного пікселя (розмір 160000).
nonprop3	double[]	Масив, що зберігає значення показника nonprop3 для кожного пікселя (розмір 160000).
pr1	double	Проміжна змінна, що використовується для обчислення співвідношення в формулі nonprop.
pr2	double	Проміжна змінна, що використовується для обчислення співвідношення в формулі nonprop.
gr	Graphics	Об'єкт для малювання зображення на формі.
mybitmap0	Bitmap	Зображення, що завантажується з файлу img1.jpg.
mybitmap1	Bitmap	Зменшена версія mybitmap0 (200x200 пікселів).
height	int	Висота зображення (200 пікселів).
width	int	Ширина зображення (200 пікселів).
i	int	Лічильник для циклу по висоті зображення.
j	int	Лічильник для циклу по ширині зображення.
q	int	Індекс, який використовується для доступу до масивів (обчислюється як $i * \text{width} + j$).
c1	Color	Колір пікселя з mybitmap1 на позиції (i, j).

a	byte	Значення альфа-каналу пікселя c1.
r	byte	Значення червоного кольору пікселя c1.
g	byte	Значення зеленого кольору пікселя c1.
b	byte	Значення синього кольору пікселя c1.
r_new	byte	Нове значення червоного кольору після обробки.
g	byte	Нове значення зеленого кольору після обробки.
b	byte	Нове значення синього кольору після обробки.

Блок-схема програми

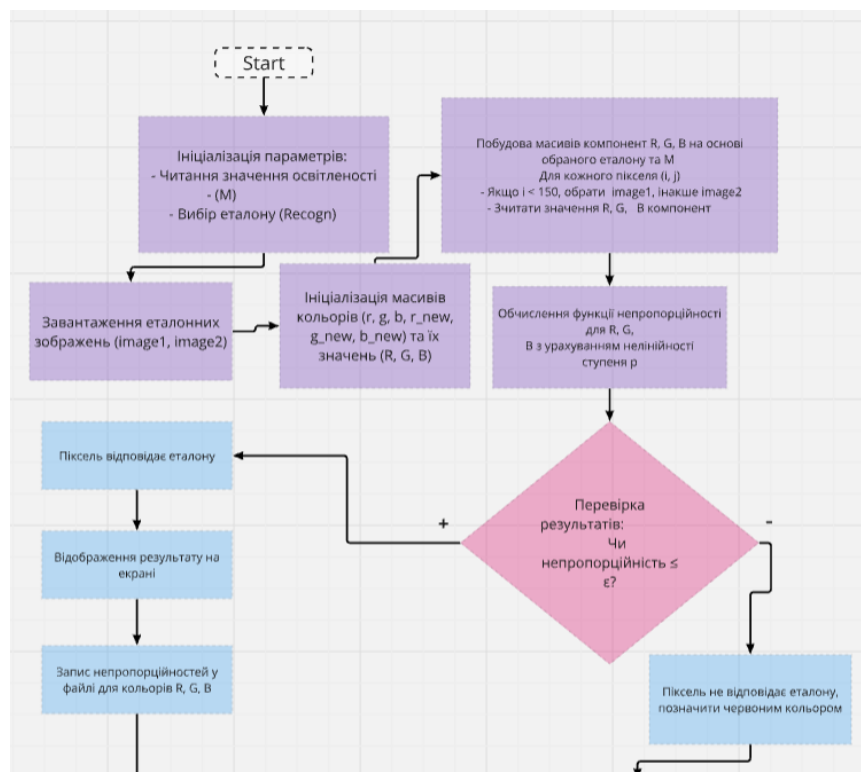


Рис 3. Блок-схема алгоритму програми.

2.6 Інструкція для користувача

Програма написана мовою C#

Ім'я проєкту – Masters

Розроблено в середовищі - Visual Studio 2019 під .NET Framework

На початку слід ввести файли еталонних зображень, та зображення для аналізу, щоб його було визнано як img1.jpg. Переконайтеся, що зображення має розмір 200 x 200 пікселів.

Запустити проєкт.

Результат роботи програми представлені в трьох вікнах:

У першому вікні (pictureBox1) буде показано вхідне зображення.

У другому вікні (pictureBox2) буде відображено спотворене зображення.

У третьому вікні (pictureBox3) буде відображено відновлене зображення.

3. КОНТРОЛЬНИЙ ПРИКЛАД

Для контрольного прикладу використовується зображення Розторопші img1.jpg . Воно завантажується із файлу, а потім піддається спотворенню, згідно наступного полінома для червоної складової пікселів:

$$r_k(q) = 0,007 R_k^3(q) + 0,007 R_k^2(q) + 0,75 R_k(q), 1 \leq q \leq N$$

Аналогічні формули застосовуються для зеленого (G) та синього (B) компонентів. У результаті кожен піксель спотворюється.

Результати роботи програми відображені в трьох рисунках.

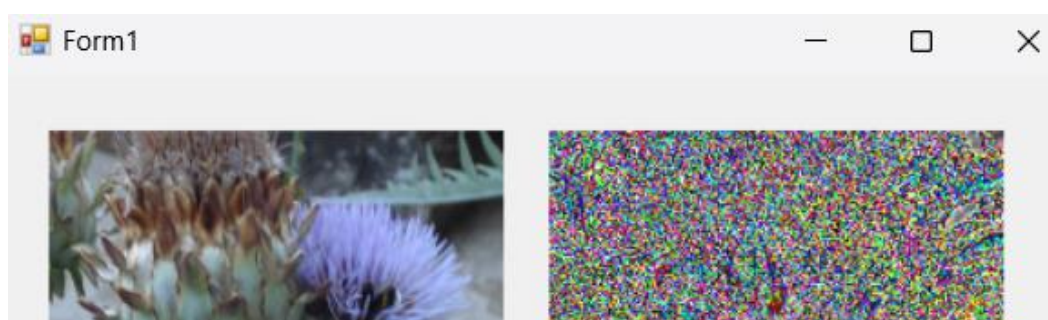


Рис 4. Результати роботи програми

- У лівому верхньому куті відображено еталонне, неспотворене зображення.
- Праворуч від нього знаходиться спотворене зображення.
- У правій нижній частині розміщено відновлене зображення, отримане шляхом розпізнавання

Отримані результати свідчать про те що незважаючи на значне спотворення зображення, яке аналізується запропонований алгоритм забезпечує успішне його розпізнавання.

ВИСНОВКИ

У рамках даної роботи було розроблено метод розпізнавання 2-D зображень, спотворених внаслідок їх обробки нелінійним об'єктом. Поставлене завдання вимагало розроблення алгоритму, який здатний розпізнати спотворене зображення.

Також було розроблено комп'ютерну програму, яка реалізує запропонований алгоритм. Це дозволяє наочно продемонструвати результати та роботу методу в реальних умовах. Робота алгоритму, та комп'ютерної програми перевірені на контрольному прикладі.

Таким чином, розроблений метод та програмне забезпечення можуть бути корисним для практичного застосування в областях, де виникають проблеми із спотворенням зображень, таких як комп'ютерна томографія, обробка зображень в оптичних системах або відеоаналітика.

СПИСОК ЛІТЕРАТУРИ

1. Gonzalez, R. C., Woods, R. E. Digital Image Processing. — Pearson, 2018.
2. Mallat, S. A Wavelet Tour of Signal Processing. — Academic Press, 2008.
3. Krizhevsky, A., Sutskever, I., Hinton, G. E. Imagenet Classification with Deep Convolutional Neural Networks // Communications of the ACM. — 2017.
4. Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints // International Journal of Computer Vision. — 2004.
5. Goodfellow, I., Bengio, Y., Courville, A. Deep Learning. — MIT Press, 2016.
6. Jain, A.K., et al.: *Machine Vision*. McGraw-Hill, 1995.

7. Kalashnykova, N., Avramenko, V. V., Kalashnikov, V., & Demianenko, V. (n.d.). On-line recognition of fragments of standard images distorted by non-linear devices and with a presence of an additive impulse interference. Unpublished manuscript, Universidad Autónoma de Nuevo León, Sumy State University, Tecnológico de Monterrey, and Central Economics & Mathematics Institute.
8. Castleman, K.R.: *Digital Image Processing*. Prentice Hall, 1996.
9. Kalashnykova, N., Avramenko, V. V., Kalashnikov, V., & Demianenko, V. (n.d.). On-line recognition of fragments of standard images distorted by non-linear devices and with a presence of an additive impulse interference. Unpublished manuscript, Universidad Autónoma de Nuevo León, Sumy State University, Tecnológico de Monterrey, and Central Economics & Mathematics Institute.
10. Voishvillo G.V. Amplification devices Textbook for High School's- M.: Radio and Communications, 1983
11. Szeliski, R.: *Computer Vision: Algorithms and Applications*. Springer, 2010.
12. Bezruk V.M., Ivanenko S.A.: Selection and recognition of the specified radio signals in the SW band. J. Information and Telecommunication Sciences, vol. 9, number 2, pp.21–25 (2018)
13. Shevchuk B. M., Zadiraka V.K., Fraier S.V., Luts V.K.: Operatyvne rozpoznavannia frahmentiv i kompleksiv syhnaliv ta vydilennia obiektiv videodanykh zasobamy obiektnykh system bezprovidnykh merezh.J. Iskusstvennyiyy intellekt, vol.3. pp.275–283 (2013)
14. Muryigin K.V.: Obnaruzhenie avtomobilnyih nomernyih znakov s ispolzovaniem predvaritelnoy obrabotki kandidatov. J. Iskusstvennyiyy intellekt, vol.3. pp.193-199 (2013)
15. Artificial Neural Networks: Concepts and Theory, IEEE Computer Society Press, 1992

16. Canny, J.: "A Computational Approach to Edge Detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
17. Karpenko, A.P.: Integral'nye harakteristiki neproporcional'nosti chislovyh funkciy i ih primenenie v diagnostike. *Vestnik SumGU*, vol. 16, pp. 20–25(2000)
18. Freeman, W.T., Garder, L., et al.: *Computational Photography: Camera, Image, and Computational Photography*. IEEE Press, 2005.
19. He, K., et al.: "Deep Residual Learning for Image Recognition". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
20. Xu, Y., et al.: "Image Recognition with Deep Neural Networks for Distorted Images". *IEEE Transactions on Image Processing*, vol. 25, pp. 3015–3026, 2016.
21. Wang, Y., et al.: "Wavelet-Based Image Enhancement for Distorted Data". *International Journal of Imaging Systems and Technology*, vol. 26, no. 2, pp. 147–157, 2016.
22. Tarel, J.P., et al.: "Recognition of Distorted 2D Images Using SIFT and HOG Descriptors". *Journal of Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 256–268, 2013.
23. Kalashnikov VV, Avramenko VV, Kalashnykova NI. Derivative disproportion functions for pattern recognition. In: Watada J, Tan SC, Vasant P, Padmanabhan E, Jain LC, editors. *Unconventional modelling, simulation, and optimization of geoscience and petroleum engineering*, Berlin-Heidelberg: Springer-Verlag; 2018, p. 95–104.
24. Kalashnikov VV, Avramenko VV, Slipushko NY, Kalashnykova NI, Konoplyanchenko AE. Identification of quasi-stationary dynamic objects with the use of derivative disproportion functions. *Procedia Comput Sci* 2017; 108(C): 2100–2109.

25. Kalashnikov VV, Avramenko VV, Kalashnykova NI, Kalashnikov Jr VV. A cryptosystem based upon sums of key functions. *Int J Comb Optim Probl Inform* 2017; 8: 31–38.
26. Kalashnykova NI, Avramenko VV, Kalashnikov VV. Sums of key functions generating cryptosystems. In: Rodriguez JMF et al., editors. *Computational Science – ICCS 2019. ICCS 2019. Lecture Notes in Computer Science*, vol 11540, Cham: Springer; 2019, p. 293–302.
27. Keith Jack Video Demystified. A Handbook for the Digital Engineer» (5th Edition), 2007.
28. Liu, C., et al.: "Super-Resolution and Image Restoration". *Springer Handbook of Image Processing*, pp. 37–58, 2014.
29. Yang, M., et al.: "Efficient Image Restoration Using Convolutional Neural Networks". *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4321–4334, 2018.

Додаток

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace Masters
{

```

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

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void button1_Click(object sender, EventArgs e)
    {
        Bitmap my = new Bitmap(200, 200);
        Bitmap my_result = new Bitmap(200, 200);
        double[] Red = new double[160000];
        double[] Red_new = new double[160000];
        double[] nonprop = new double[160000];
        double[] nonprop2 = new double[160000];
        double[] nonprop3 = new double[160000];

        double pr1, pr2;

        // Завантажити еталонне зображення
        Bitmap mybitmap0 = new Bitmap("C:\\Users\\kitKa\\OneDrive\\Робочий
стіл\\Masters\\Masters\\bin\\Debug\\img1.jpg");
        Bitmap mybitmap1 = new Bitmap(mybitmap0, new Size(200, 200));

        // Встановлюємо еталонне зображення в pictureBox1
    }
}
```

```

pictureBox1.Image = mybitmap1;

int height = 200;
int width = 200;

// Обробка зображення
for (int i = 0; i < height; i++)
    for (int j = 0; j < width; j++)
    {
        int q = i * width + j;
        Color c1 = mybitmap1.GetPixel(i, j);

        byte a = c1.A;
        byte r = c1.R;
        Red[q] = (double)r;
        byte g = c1.G;
        byte b = c1.B;

        double rr = 0.001 * Math.Pow(Red[q], 3) + 0.001 * Math.Pow(Red[q],
2) + 0.95 * Red[q];
        Red_new[q] = rr;
        byte r_new = (byte)rr;

        double gg = 0.001 * Math.Pow(g, 3) + 0.001 * Math.Pow(g, 2) + 0.95
* g;
        g = (byte)gg;

        double bb = 0.001 * Math.Pow(b, 3) + 0.001 * Math.Pow(b, 2) + 0.95
* b;

```

```

    b = (byte)bb;

    Color c2 = Color.FromArgb(a, r_new, g, b);
    my.SetPixel(i, j, c2);
}

// Встановлюємо спотворене зображення в pictureBox2

pictureBox2.Image = my;

// Тут ваш код для обчислення nonprop, nonprop2 і nonprop3
// ***** Обчислення nonprop
for (int i = 0; i < height - 1; i++)
    for (int j = 1; j < width - 1; j++)
    {
        int q = i * height + j;
        chisl = (Red_new[q - 1] + Red_new[q]);
        znam = (Red[q - 1] + Red[q]);
// if (Math.Abs(chisl) <= 1e-4 && Math.Abs(znam) <= 1e-4 &&
Math.Abs(Red_new[q]) <= 1e-4 && Math.Abs(Red[q]) <= 1e-4)

        // nonprop[q] = 0;

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

            pr1 = chisl / znam;
            pr2 = Red_new[q] / Red[q];

```

```

        nonprop[q] = pr1 - pr2;
    }
    else
        nonprop[q] = 1000;
    // sw.WriteLine(" q={0}    nonprop={1}", q, nonprop[q]);
}

//*****
//***** Обчислення nonprop2
for (int i = 0; i < height - 1; i++)
    for (int j = 1; j < width - 1; j++)
    {
        int q = i * height + j;
        chisl = (nonprop[q - 1] + nonprop[q]);
        znam = (Red[q - 1] + Red[q]);
// if (Math.Abs(chisl) <=1e-4 && Math.Abs(znam) <=1e-4 &&
Math.Abs(Red_new[q])<=1e-4 && Math.Abs(Red[q])<=1e-4)

        // nonprop2[q] = 0;

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

            pr1 = chisl / znam;
            pr2 = nonprop[q] / Red[q];

            nonprop2[q] = pr1 - pr2;
        }
        else
            nonprop2[q] = 1000;

```

```

// sw.WriteLine(" q={0}    nonprop2={1}", q, nonprop2[q]);

}

//*****
//*****Обчислення nonprop3
for (int i = 0; i < height - 1; i++)
    for (int j = 1; j < width - 1; j++)
        {
            int q = i * height + j;
            chisl = (nonprop2[q - 1] + nonprop2[q]);
            znam = (Red[q - 1] + Red[q]);
            // if (Math.Abs(chisl) <=1e-4 && Math.Abs(znam) <=1e-4 &&
Math.Abs(Red_new[q])<=1e-4 && Math.Abs(Red[q])<=1e-4)
                // nonprop3[q] = 0;

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

                    pr1 = chisl / znam;
                    pr2 = nonprop2[q] / Red[q];

                    nonprop3[q] = pr1 - pr2;
                }
            else
                nonprop3[q] = 1000;
            sw.WriteLine(" q={0}    nonprop3={1}", q, nonprop3[q]);
        }

//*****

```



```

Color c3 = new Color(); ;
for (int i = 0; i < height; i++)
    for (int j = 0; j < width; j++)
        {
            int q = i * width + j;
            // nonprop3[q] = 0;
            if (Math.Abs(nonprop3[q])<=1e-4)

                c3 = mybitmap1.GetPixel(i, j);

            /*  c3 = etalon1.GetPixel(i, j);

else

            c3 = etalon2.GetPixel(i + 300, j + 300);*/

my_result.SetPixel(i, j, c3);

// Восстановление изображения
Color c3 = new Color();
for (int i = 0; i < height; i++)
    for (int j = 0; j < width; j++)
        {
            int q = i * width + j;
            if (Math.Abs(nonprop3[q]) <= 1e-4)
                c3 = mybitmap1.GetPixel(i, j);
            my_result.SetPixel(i, j, c3);
        }

```

```
// Встановлюємо відновлене зображення в pictureBox3  
pictureBox3.Image = my_result;  
}  
}  
}
```