

## ДОСЛІДЖЕННЯ МЕТОДІВ МАСШТАБУВАННЯ ЦИФРОВИХ ЗОБРАЖЕНЬ

**В. В. Стецюк**

здобувач вищої освіти першого (бакалаврського) рівня, 2 курс,  
спеціальність «Комп'ютерні науки», навчально-науковий інститут кібернетики  
інформаційних технологій та інженерії

Науковий керівник – к.т.н., доцент О. Р. Мічута

*Національний університет водного господарства та природокористування,  
м. Рівне, Україна*

У статті розглянуто основні та найпоширеніші методи масштабування та фільтрації цифрових зображень, визначені їх переваги та недоліки. Дослідженню підлягають методи інтерполяції, включно з сплайн-інтерполяцією, фільтрації та сучасні підходи з використанням нейронних мереж. Ці методи використовуються для зміни роздільної здатності зображення, зберігаючи при цьому якість.

**Ключові слова:** інтерполяція, сплайн-інтерполяція, фільтр, нейронна мережа, зображення.

The article discusses the fundamental and most common methods of scaling and filtering digital images, their advantages and disadvantages are identified. Interpolation methods, including spline interpolation, filtering, and modern approaches using neural networks are subject to study. These methods are used to change the resolution of an image while maintaining its quality.

**Keywords:** interpolation, spline interpolation, filter, neural network, image.

**Цифрове зображення** – це матриця пікселів. Піксель є найменшим елементом, що формує зображення. Кожен піксель містить інформацію про колір, яка зазвичай виражається через кольорові компоненти, такі як червоний, зелений та синій (RGB). Роздільна здатність зображення визначається кількістю пікселів по горизонталі та вертикалі, що безпосередньо впливає на деталізацію зображення.

Масштабування зображень є актуальною задачею сьогодення, адже втрата якості може трапитись під час обробки або передачі файлів через мережу чи від самого початку бути нечіткою. На цей момент існує багато алгоритмів масштабування зображень, кожен з яких має свої переваги і недоліки.

Методи масштабування дозволяють змінити роздільну здатність зображення з максимально можливим збереженням якості, а методи фільтрації ефективно прибирати пікселізацію та розмиття. Вони мають широке застосування в комп'ютерній графіці, обробці відео, супутниковому зв'язку.

Збереження якості зображення при масштабуванні є важливим для аналізу фізичних полів невеликої території на основі цифрових супутникових зображень, які фіксують велику площу поверхні. Дозволяє економити ресурси на передачу високо інформативних даних.

**Мета статті** – дослідити та зробити порівняльний аналіз основних методів масштабування та фільтрації цифрових зображень, зокрема традиційних інтерполяційних підходів і сучасних методів на основі нейронних мереж, для визначення їх ефективності, переваг і недоліків при зміні роздільної здатності зображень.

**Цифрова інтерполяція зображень** – це процес створення неперервної поверхні інтенсивності з дискретних даних зображення зразків зображення. Існує багато різних методів інтерполяції, кожен з яких призводить до різного вигляду кінцевого зображення.

**Найближчий сусід.** Цей метод є найпростішим методом масштабування зображень. Ідея даного алгоритму полягає в створенні нової решітки, коли ми працюємо над масштабуванням зображення [2]. Для кожного пікселя на новій сітці ми шукаємо найближчий піксель на старій та призначаємо його значення новому пікселю, тобто метод вибирає найближчий піксель як репрезентативний (рис. 1). Значним недоліком методу є значна втрата якості зображення.

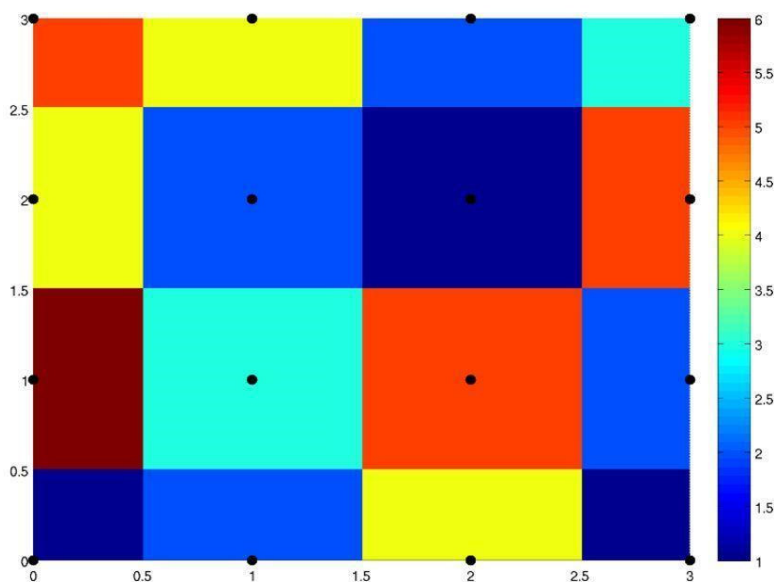


Рис. 1. Інтерполяція методом найближчого сусіда

**Білінійна інтерполяція.** Цей метод враховує значення чотирьох найближчих пікселів з оригінального зображення, щоб обчислити значення кожного нового пікселя у масштабованому зображенні. Спочатку розташування пікселя, яке називається Р, у збільшеному зображенні зазнає трансформації у відповідне місце на вихідному зображенні [1]. Після цього оцінюється вплив квартету піксельних точок, а саме А, В, С і D (рис. 2). Ця оцінка базується на принципі, що близькість до точки Р безпосередньо впливає на обчислене значення; чим ближча відстань, тим сильніший вплив, що означає підвищену вагомість. Розглянемо координати А, В, С і D як  $(i, j)$ ,  $(i, j + 1)$ ,  $(i + 1, j)$  та  $(i + 1, j + 1)$  відповідно. Координати Р позначаються як  $(u, v)$ . Алгоритм білінійної інтерполяції для пікселя Р проходить в три окремі етапи:

Крок 1: обчислити вплив А і В і позначити його як Е.

$$f(i, j + v) = [f(i, j + 1) - f(i, j)] v + f(i, j). \quad (1)$$

Крок 2: розрахувати вплив С і D і позначити його як F.

$$f(i + u, j + v) = [f(i + 1, j + 1) - f(i + 1, j)] v + f(i + 1, j). \quad (2)$$

Крок 3: розрахувати вплив Е і F і позначити його як Р.

$$f(i + u, j + v) = (1 - u)(1 - v)f(i, j) - (1 - u)vf(i, j + 1) + u(1 - v)f(i + 1, j) + uvf(i + 1)(j + 1). \quad (3)$$

|   |  |  |   |   |
|---|--|--|---|---|
| A |  |  | E | B |
|   |  |  |   |   |
|   |  |  |   |   |
|   |  |  | P |   |
| C |  |  | F | D |

Рис. 2. Схема білінійної інтерполяції

Головною перевагою даного методу є забезпечення плавних переходів, а недоліком поява розмиття.

**Бікубічна інтерполяція.** В цьому методі навколо обраного пікселя використовується  $4 \times 4$  сітка для обчислення його значення. Мета полягає в генерації кубічної поліноміальної функції, яка плавно проходить через ці 16 сусідніх пікселів, забезпечуючи безперервність і мінімізуючи різкі зміни в оцінюваних значеннях [1]. Для досягнення цього бікубічна інтерполяція застосовує наступні кроки (рис. 3):

Крок 1: Враховуючи основне розташування  $(u, v)$ , де потрібно оцінити значення пікселя, визначається найближча сітка сусідніх пікселів  $4 \times 4$ . Ця сітка центрується навколо точки  $(u, v)$ .

Крок 2: Бікубічна інтерполяція обчислює середньозважене значення 16 сусідніх пікселів. Вага, що присвоюється кожному пікселю, залежить від відстані між цільовим положенням і положенням сусіднього пікселя. Пікселі, розташовані ближче до цільового положення, мають більшу вагу в процесі усереднення.

Крок 3: Після встановлення функції кубічного полінома значення пікселя в цільовому розташуванні визначається шляхом обчислення полінома за потрібними координатами  $(v, u)$ .

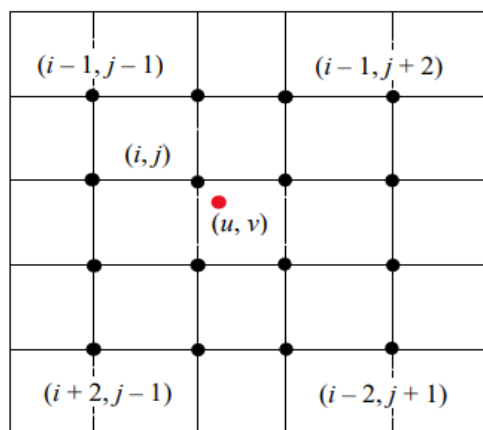


Рис. 3. Схема бікубічної інтерполяції

Цей метод забезпечує кращу деталізацію ніж білінійна інтерполяція, але й потребує потужніших ресурсів через його складність.

**Інтерполяція сплайнами.** Ідея полягає в побудові поліномів між парами сусідніх вузлів інтерполяції, причому для кожної пари вузлів будується свій поліном. Сплайн-функцією або сплайном називають кусково-поліноміальну функцію, що визначена на відрізку і має на цьому відрізку деяке число безперервних похідних [6]. Слово «сплайн» (англійське «spline») означає гнучку лінійку, що використовується для проведення гладких кривих через задані точки площини. Основною перевагою сплайнів є можливість локально змінювати форму кривої на виділеному проміжку значень.

**Кубічний сплайн.** Алгоритм розрахунку коефіцієнтів сплайну та їх використання є достатньо громіздким і складається з таких етапів: формування вихідної інформації (визначення вузлів та значень функції у вузлах); побудова системи рівнянь, яка враховує умови стосовно похідних; розв'язання побудованої системи рівнянь – визначення коефіцієнтів сплайну; обчислення значень сплайну у проміжних точках [9]. Для побудови необхідно побудувати  $n$  многочленів третьої степені.

**В-сплайн.** В-сплайн – це сплайн-функція, що має мінімальний носій для заданого степеня, гладкості та області визначення [4]. Фундаментальна теорема стверджує, що довільна сплайн-функція заданого степеня, гладкості і області визначення може бути представлена як лінійна комбінація В-сплайнів того ж ступеня і гладкості на тій самій області визначення. Термін В-сплайн запровадив Ісак Яков Шонберг в 1947 році.

**Фільтрація цифрових зображень.** Фільтрація – один із найважливіших процесів в обробці зображень. Вона дозволяє видалити небажаний шум та згладити гострі краї, зберігаючи важливі деталі. Вибір методу фільтрації залежить від типу спотворення та обчислювальних можливостей. Методи фільтрації можуть застосовуватись до і після масштабування. Перед масштабуванням щоб небажані деталі не викривляли зображення надалі. Після, щоб прибрати спотворення, які залишили методи масштабування.

**Фільтр Ланцоша.** Фільтр використовує функцію:

$$L(x) = (x), \quad (4)$$

яка базується на функції sinc, для визначення значення нових пікселів при збільшенні розміру зображення. Функція sinc визначається як:

$$(x) = \frac{\sin \sin (\pi x)}{\pi x}. \quad (5)$$

Фільтр має параметр  $\alpha$ , який визначає розмір вікна фільтра. Більший параметр  $\alpha$  забезпечує краще збереження деталей і гостроти країв, але також може призвести до з'явлення артефактів у вигляді коливань навколо гострих країв [3]. Функція Ланцоша є обмеженою версією функції sinc, і вона дозволяє зменшити обчислювальну складність, обмежуючи кількість використовуваних сусідніх пікселів. Для кожного нового пікселя обчислюється зважене середнє значення сусідніх пікселів оригінального зображення. Ваги визначаються функцією Ланцоша, яка залежить від відстані між новим пікселем і сусідніми пікселями.

**Гауссівське розмиття.** Воно ґрунтується на гауссівській функції, що являє собою розподіл інтенсивності значень пікселів [10]. Гауссівське розмиття використовує цей розподіл, щоб згладити грані та розрізнити деталі, що сприяє підвищенню різкості зображення та видаленню розмитості. Розшарування контурів має два основних параметри:  $kCInputRadiusKey$  (визначає, як далеко залучені пікселі) та  $kCInputIntensityKey$  (контролює ступінь поліпшення якості контурів та деталей). Розшарування контурів використовує внутрішній оператор розмиття для створення розмитої версії зображення. Надалі від цієї розмитої версії віднімається оригінальне зображення, що призводить до виділення контурів і деталей. Гауссівським розмиттям можна скористатись для видалення небажаних деталей або розмиття чутливої інформації на зображенні. 2D рівняння Гауса (рис. 4):

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (6)$$

де  $\sigma$  – стандартне відхилення розподілу, а  $x, y$  – координати в просторі ядра фільтра.

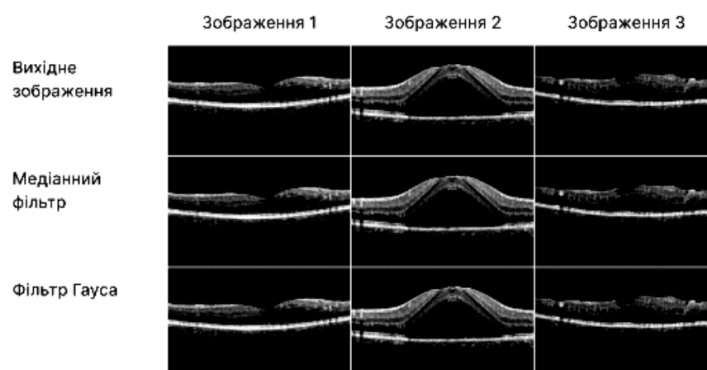


Рис. 4. Розмивання Гауса та медіанна фільтрація

**Медіанний фільтр.** В основі даного алгоритму полягає корекція значень кольору та яскравості кожного пікселя з урахуванням аналогічних значень сусідніх пікселів, що допомагає знизити так звану «зернистість» зображення, яка суттєво заважає алгоритмам цифрової обробки зображень [5] (рис. 5). Медіанний фільтр є ефективним проти імпульсних перешкод (рис. 4).

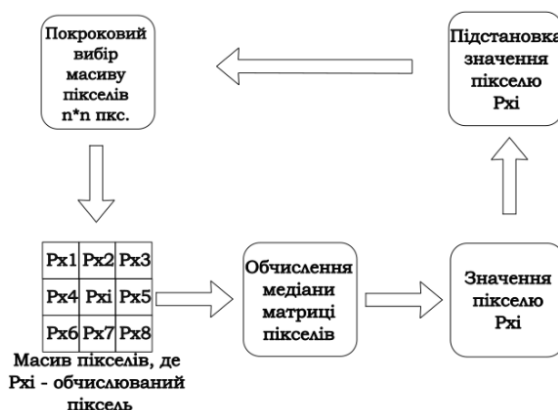


Рис. 5. Схематичне зображення алгоритму роботи медіанного фільтру

**Фільтр середнього значення.** Ідея фільтрації частково схожа з ідеєю медіанної фільтрації, але замість використання значення медіани з упорядкованого списку обраховують середнє значення сусідів, включно із обраним пікселем. Це призводить до усунення значень пікселів, які не є репрезентативними для їхнього оточення.

**Адаптивний фільтр.** Цей тип фільтрів дозволяє налаштувати параметри фільтру залежно від характеристик шуму і сигналу. Зберігає ключові деталі зображення та ефективно прибирає шуми. Використовують для відновлення зображень, що зазнали дії імпульсного шуму, гаусового шуму або їх комбінацій. Ефективно працює в задачах, де потрібно зберігати важливі структурні деталі, наприклад, межі об'єктів або текстури.

**Нейронні мережі.** На Python існує велика кількість бібліотек, а отже, постає задача вибору оптимальної для конкретних задач проєкту. Перевагами використання нейронних мереж в обробці цифрових зображень є висока адаптивність та якість, але через складність використання виникає потреба у великих ресурсах.

**Keras.** API для побудови нейронних мереж на високому рівні. Keras призначений для прискорення створення моделі нейромережі. Він здійснює модульне виконання: модель розглядається як послідовність чи як граф автономних, повністю сконфігурованих модулів, які можна з'єднувати разом із мінімальними обмеженнями [7]. Нові модулі легко додаються у вигляді нових класів та функцій.

**TensorFlow.** Бібліотека розроблена Google. Фреймворк працює із статичними графами, має підтримку Python, C++ та R, надає API для .NET, Java, C, Go, JavaScript та доступний для стаціонарних та мобільних пристроїв [7]. TensorFlow – фреймворк із низьким рівнем програмування. Основним недоліком є складність використання низькорівневого API налаштування статичних графів обчислень.

**PyTorch.** Фреймворк використовує динамічний обчислювальний граф, який передбачає побудову нового графа на кожному прохідному етапі [7]. Простіша у використанні бібліотека, проте програє у випадку використання для великих систем та має менше специфічних інструментів, ніж TensorFlow.

**Apache MXNet.** Бібліотека підтримується компаніями Amazon, Microsoft [7]. Фреймворк забезпечує високу продуктивність і відмінну масштабованість, має гарні навички класифікації. Незважаючи на ці переваги, MXNet має невелику сферу застосування та дуже обмежену до використання документацію. Підтримка різних мов програмування: Python, Scala, R, JavaScript, Julia, Perl, та C++.

**Topaz Gigapixel AI.** В основі алгоритмів програми лежить цілий комплекс нейромереж [8]. Програма розбирає фото на пікселі і сортує їх за кольором, потім формується маска в потрібній роздільній здатності і далі система схематично додає пікселі потрібних кольорів на наявне зображення, таким чином буквально «домальовуючи» його до необхідного покращення. Topaz Gigapixel AI не просто «розтягує» зображення, а розпізнає деталі і текстури, роблячи їх більш чіткими. Ця програма є комерційним продуктом, для використання якого потрібно придбати ліцензію.

**Висновки.** Існує безліч методів масштабування зображень і кожен з них має свої переваги та недоліки. Вибір методу залежить від типу задачі, наявності потужних ресурсів для обробки та вимог до результату. Існують традиційні (ручні) способи, які можуть бути корисними у випадках відсутності потужних обчислювальних ресурсів або потреби контролю процесів. Методи із застосуванням нейронних мереж можуть мати високу точність, але водночас бути ресурсозатратними. Вибір оптимального ресурсу для забезпечення цілісності даних при масштабуванні та затрат залишається за користувачем.

1. Бодашевський Д. Р., Потапова К. Р. Аналіз принципів підвищення якості зображень через призму бікубічної інтерполяції та згорткових нейронних мереж. *Інтернаука*. 2023. № 15. С. 48–53. URL: [http://irbis-nbuv.gov.ua/cgi-bin/irbis\\_nbuv/cgiirbis\\_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE\\_FILE\\_DOWNLOAD=1&image\\_file\\_name=PDF/mnj\\_2023\\_15\\_9.pdf](http://irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE_FILE_DOWNLOAD=1&image_file_name=PDF/mnj_2023_15_9.pdf) (дата звернення: 23.03.2025).
2. Масштабування зображень. URL: <https://pzs.dstu.dp.ua/ComputerGraphics/scaling/index.html> (дата звернення: 23.03.2025).
3. Бодашевський Д. Р., Потапова К. Р. Метод збільшення розміру зображень на основі бікубічної інтерполяції та фільтра Ланцоша. КІП ім. Ігоря Сікорського. С. 314–315.
4. Інтерполяція. studfile. URL: <https://studfile.net/preview/9721094/> (дата звернення: 28.02.2025).
5. А. Щербатюк, С. Тужанський. Методи оптичної когерентної томографії та алгоритми фільтрації зображень для офтальмологічної діагностики, *Оптим.-ел. інф.-енерг. техн.* Лип 2024. Вип. 47, Вип. 1. С. 148–154. URL: <https://oeipt.vntu.edu.ua/index.php/oeipt/article/view/700/651> (дата звернення: 07.04.2025).
6. Сплайн-інтерполяція. URL: [https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj\\_komp%27yuterne\\_modelyuvannya\\_system\\_procesiv/t1/612..htm](https://web.posibnyky.vntu.edu.ua/fksa/2kvetnyj_komp%27yuterne_modelyuvannya_system_procesiv/t1/612..htm) (дата звернення: 01.03.2025).
7. Романчук М. П. Обґрунтування типу фреймворків глибокого навчання для оброблення даних дистанційного зондування землі. *Збірник наукових праць ЖВІ*. 2019. № 16. С. 73.
8. Тулашвілі Ю. Й., Лук'янчук Ю. А. Використання програмного забезпечення на основі штучного інтелекту для обробки зображень. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2021. № 43. С. 221–222.
9. Норік Л. О. Сплайн-інтерполяція в аналізі результатів економічної діяльності. *Сучасні тенденції розвитку фінансових та інноваційно-інвестиційних процесів в Україні* : зб. матеріалів VI Міжнародної науково-практичної конференції, 2–3 березня 2023 р. Вінниця : ВНТУ, 2023. С. 398–400.
10. Яркул Р. І., Парамуд Я. С. Особливості алгоритмічних та програмних засобів опрацювання нечітких зображень. *Computer systems and networks*. 2023. Vol. 5, No. 1. С. 175. URL: [https://science.lpnu.ua/uk/csn/vsi-vypusky/vypusk-5-nomer-1-2023/osoblyvosti-algorytmichnyh-ta-programnyh-zasobiv-opracyuvannya?utm\\_source=chatgpt.com](https://science.lpnu.ua/uk/csn/vsi-vypusky/vypusk-5-nomer-1-2023/osoblyvosti-algorytmichnyh-ta-programnyh-zasobiv-opracyuvannya?utm_source=chatgpt.com) (дата звернення: 07.04.2025).