

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Кваліфікаційна робота

освітній ступінь – бакалавр

на тему: «Застосування полів Гіббса у розпізнаванні образів»

Виконав: студент 4-го року навчання
освітньої програми «Прикладна
математика», спеціальності 113
Прикладна математика

Сичов Володимир Сергійович

Керівник: Михалевич В.М.,
Доцент

Рецензент _____

(прізвище та ініціали)

Кваліфікаційна робота захищена

з оцінкою _____

Секретар ЕК _____

«____» _____ 20____ р.

Київ – 2021

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1: Загальний огляд задачі обробки зображень	4
1.1: Загальний огляд.....	4
1.2: Можливості прикладного застосування.....	4
РОЗДІЛ 2: Основні ідеї математичного моделювання з використанням Баєсівського підходу.....	6
2.1: Баєсівська статистика в обробці зображень.....	6
2.2: Що таке алгоритм семплування?.....	7
2.3: Моделювання з використанням баєсівської статистики.....	7
2.4: Випадкове поле маркова.....	8
2.5: Представлення зображень як бінарної моделі.....	12
2.6: Модель Ізінга.....	15
2.7: Алгоритм семплуванні Гіббса.....	17
2.8: Шум у зображеннях.....	18
2.9: Тип шуму «Сіль-перець».....	19
РОЗДІЛ 3: Реалізація алгоритму обробки зображень.....	20
Висновок:	22
Додаток:.....	23

Вступ

Актуальність теми

В наш час використання цифрових зображень набуло широко застосування в багатьох сферах, зокрема в комунікації (супутникове телебачення), медицині (магнітно-резонансна томографія, комп'ютерна томографія) а також в багатьох інших сферах технологічних досліджень. Дані в цих сферах отримуються за допомогою датчиків, які зчитують інформацію з навколишнього світу та представляють її у оцифрованому вигляді, а саме як цифрові зображення. Проте нерідко цифрові зображення мають шуми, які ускладнюють їхній аналіз і подальше використання. На наявність шумів впливає велика кількість чинників: несправності в обладнанні, проблеми під час передачі даних, вплив природних чинників. Крім цього, шуми можуть утворюватися внаслідок передачі або стиснення даних. Таким чином, позбавлення зображення від шуму є актуальною задачею.

В обробці зображень, задача відновлення зображень є фундаментальною задачею, тому існує необхідність в знаходженні методів ефективного її вирішення. Задачі відновлення зображень мають два основні підходи: *частотний* і *просторовий*. Дана робота приділяє увагу з підмножиною частотних методів, а саме статистичним.[5]

Мета дослідження: зображення байєсівських підходів в задачах цифрової обробки зображень та у відновленні зашумлених зображень за допомогою алгоритму семплування Гіббса.

Об'єкт дослідження: чорно-білі зображення, які представлені з використанням моделі Ізінга.

Предмет дослідження: алгоритм для знаходження апостеріорного розподілу Гіббса в зашумлених зображеннях.

РОЗДІЛ 1: Загальний огляд задачі обробки зображень

1.1: Загальний огляд

Є два основні типи задач цифрової обробки зображень:

Низькорівневі – задачі, які на основі локальних дій намагаються покращити якість зображення (фільтри або позбавлення шуму);

Високорівневі – задачі ідентифікації та знаходження об'єктів;

Однією зі спільних рис цих типів задач являється представлення постановки задачі як композиції функції від зображення та шуму:

$$\text{Сигнал} = f(\text{Зображення}) \otimes \text{Шум} \quad (1)$$

де $\otimes$ - деякий оператор комбінування,

f – функція, яка позначає те, що сигнал приходить у деякому вигляді, відмінному від оригінального зображення.

Дана робота присвячена задачі низькорівневого типу, а саме відновленню зашумлених зображень.[5]

1.2: Можливості прикладного застосування

З розвитком технологій, кількість сфер, в яких можна використати цифрову обробку зображень постійно збільшується. Однією з таких сфер є медична сфера, а саме томографія.

Як відомо, зображення в томографії ніколи не були чіткими, через що діагностування захворювань ускладнюється. Через шуми на зображеннях і необхідність в правильному діагнозі доводиться розширювати спектр аналізів і процедур необхідних для коректного визначення причини хвороби, що в свою чергу збільшує витрати ресурсів на одного пацієнта та збільшує навантаження на здоров'я.

Однією зі спроб вирішити цю проблему є використання штучного інтелекту для діагностування хвороби, проте шуми на зображеннях можуть знизити

точність діагнозу, що в свою чергу може привести до летальних наслідків.
Тому задача позбавлення шумів є актуальною для галузі медичних послуг.

РОЗДІЛ 2: Основні ідеї математичного моделювання з використанням Баєсівського підходу

2.1: Баєсівська статистика в обробці зображень

Для початку, розглянемо, якою ж є роль байєсівської статистики в області аналізу та обробки зображень. Визначимо її загальну мету та розглянемо методи.

З курсу теорії ймовірності нам відомо, що основна мета байєсівського виводу полягає в знаходженні повного апостеріорного розподілу певної множини випадкових величин. Якщо розглядати це в контексті цифрової обробки зображень, то це означає, що апостеріорний розподіл узагальнює знання про справжнє зображення на основі апріорного та даних спостереження. Таким чином, за можливості надання вибірок з апостеріорного розподілу є можливість їхнього використання для статистичного виведення про справжнє зображення.[1]

Проте, отримання такого розподілу дуже часто не є тривіальною задачею, оскільки часто вимагає обчислення інтегралів, які є складно обчислювані для нетривіальних моделей. Тому в таких випадках, нам на допомогу приходить сімейство методів симуляції під назвою методи Монте-Карло, а точніше методи Монте-Карло з марковськими ланцюгами (англомовна абревіатура для позначення даного сімейства методів – МСМС), завдання яких полягає в моделюванні невідомого ймовірнісного розподілу. Генерація даного розподілу відбувається за допомогою ергодичних ланцюгів Маркова, в яких необхідний нам розподіл є стаціонарним.[1]

Крім цього, за допомогою ланцюга Маркова може бути представлена фізична система, яка описує еволюцію стану та задовольняє певну локальну характеристику. В подальшому, ми розглянемо модель полів Маркова і їхню роль в байєсівському підході та методах Монте-Карло на марковських ланцюгах для семпсування розподілу.[1]

2.2: Що таке алгоритм семплування?

Основною ідеєю алгоритму семплування є апроксимація деякого, складно обчислюваного інтегралу за допомогою взяття вибірок з деякого розподілу. Саме семплування означає, що нам потрібно знайти деякий граничний розподіл.

В основі ідеї семплування Монте-Карло на ланцюгах Маркова є те, що існує можлива оцінка для будь-якого математичного сподівання завдяки ергодичним середнім. Іншими словами, ми можемо обчислити статистику апостеріорного розподілу, якщо ми маємо N вибірок з даного розподілу, які були обраховані за допомогою такої симуляції:

$$E[f(s)]_P \approx \frac{1}{N} \sum_{i=1}^N f(s^{(i)}) \quad (2)$$

P – апостеріорний розподіл, який необхідно знайти;

$f(s)$ – математичне сподівання;

$s^{(i)}$ – i -та симуляція вибірки з розподілу P .

В цифровій обробці зображень є три основні типи задач, які реалізуються за допомогою баєсівської статистики:

1. Задача структурованого моделювання шуму;
2. Задача відновлення пошкоджених даних за допомогою апріорних розподілів;
3. Задача класифікації.[3]

2.3: Моделювання з використанням баєсівської статистики

Нехай x – деяке зображення, y – відповідний йому сигнал (певним чином спотворений). Як правило, y представляється як ґратка з дискретних значень.

Індивідуальний компонент сигналу y називається пікселем.

Згідно з теоремою Байєса, ми можемо записати умовний розподіл таким чином:

$$\pi(x|y) = \frac{\pi(y|x)\pi(x)}{\pi(y)} \propto \pi(y|x)\pi(x) \quad (3)$$

$\pi(x|y)$ – функція правдоподібності, завданням якої є опис процесу формування даних для окремого оригінального зображення;
 $\pi(x)$ – апіорна ймовірність, описує будь-які апіорні знання про властивості оригінального зображення.[1]

Розподіл $\pi(y)$ недостатньо інформативний по відношенню до зображення x , тому що нас, перш за все, цікавить можливість робити певні висновки стосовно зображення x , беручи до уваги апостеріорний розподіл $\pi(x|y)$. Як не дивно, але щоб це зробити, потрібно знайти деякі компоненти моделі. Перше, що необхідно знайти, це придатне представлення для x . Придатність представлення залежить від того, яку мету ми переслідуюмо при обробці зображення. Наступне, що нам потрібно зробити, це вирішити, які апостеріорні та апіорні моделі ми будемо використовувати в нашій роботі. Зосередимся на можливостях моделювання.[1]

2.4: Випадкове поле маркова

Почнемо з моделей, в яких x та y можуть бути представлені набором значень пікселів.

Нехай I – скінченна та зліченна множина, яка являє собою вузли зображення. Ці вузли будемо називати пікселями. Для визначення моделі, в основі якої будуть використовуватися пікселі, нам необхідно звернути увагу на клас випадкових полів Маркова.

Для початку, введемо відношення сусідства $\sim$, заданого на множині I . Дане відношення задається наступним чином:

Нехай i та j – пікселі з множини I , $i \in I$, $j \in I$;

1. Якщо i є сусідом j : $i \sim j$, то й j є сусідом i : $j \sim i$. (Відношення є симетричним на I).

2. Прийнято вважати, що i не може бути сусідом сам для себе. (Відношення не є рефлексивним на I)

Введемо наступні означення:

Означення: Випадкове поле – це множина, яка складається з набору випадкових значень(змінних), таких, що $\{x_i: i \in I\}$ і таких, що кожна змінна x_i набуває значень зі скінченної множини X . Множина сусідів для вузла i буде позначатися ∂i , підмножина вузлів $A \subseteq I$ буде позначатися як, $x_A = \{x_i: i \in A\}$ та $x_{-A} = \{x_i: i \in I \setminus A\}$.

Означення: x конфігурація – це елемент з $X^{|I|}$.

Означення: Множина вузлів називається клікою C , якщо вона задається наступним чином:

якщо $x_i, x_j \in C$, то тоді x_i належить околу x_j ($x_j \in \partial i$);

якщо $x_i, x_j \in C$, то тоді x_j належить околу x_i ($x_i \in \partial j$);

Означення: Випадкове поле називається випадковим полем Маркова, якщо умовний розподіл будь-якого пікселі, при умові наявності інших пікселів, залежить лише від значень пікселів, які є сусідами з даним пікселем. Іншими словами можна сказати, що:

$$\pi(x_i | x_{-i}) = \pi(x_i | x_{\partial i}) \quad (4)$$

Умова (4) називається локальною характеристикою випадкового поля.[1]

У цифровій обробці зображень, є три найпоширеніші структури для систем околу, які складаються з чотирьох, восьми та дванадцяти найближчих сусідів.

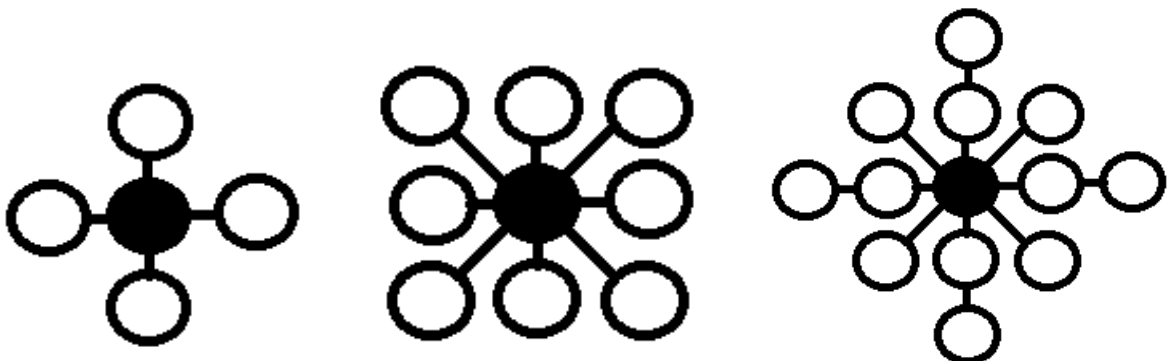


Рис.1

Рис.2

Рис.3

Рис.1 – система околу, яка складається з чотирьох сусідів;

Рис.2 – система околу, яка складається з восьми сусідів;

Рис.3 – система околу, яка складається з дванадцяти сусідів;

В цифровій обробці зображень найбільш використовуваною є система околу, яка складається з чотирьох сусідів. В реалізації даної роботи використовується система, яка складається з чотирьох сусідів.[1]

Випадкові поля маркова є дуже важливі у цифровій обробці зображень. Чому так? Є дві основні причини:

1. Дуже складно змодельовати сумісний розподіл для зображення x на решітці I , тому що не завжди зрозуміло, з чого потрібно починати моделювання цього розподілу. Для розбиття задачі на більш керовані підзадачі використовують повні умовні ймовірності. Тобто припущення про поведінку пікселя x_i буде базуватися на основі знання про конфігурації сусідів пікселя x_i . [3]
2. Зв'язок, який існує між методами Монте-Карло на ланцюгах Маркова та випадковими полями Маркова полягає в тому, що схеми оновлення, які використовуються в методах Монте-Карло, вимагають обчислення локальних повних ймовірностей(4). Припустимо, що кількість сусідів є значно меншою, ніж $n = |I|$ (така ситуація складається у більшості прикладів). Тоді при використанні семплування Гіббса, наприклад, для проходження по зображенню потрібно $O(n)$ операцій, за умови використання випадкового поля Маркова, тим часом як для того ж семплування Гіббса але на випадковому полі, яке не є марковським, необхідно $O(n^2)$ операцій для повного проходження по зображенню. [3]

Тепер, після того, як ми визначилися з моделюванням, у нас постає одне питання:

Чи існує граничний сумісний розподіл і за яких умов, якщо ми маємо набір локальних характеристик? Як знайти цей розподіл? Чи є цей розподіл унікальним? Для того, щоб дізнатися відповіді на ці питання, нам потрібно розглянути теорему Гаммерлі-Кліффорда, яка стверджує, що множина, утворена повними умовними ймовірностями, визначає щільність сумісного розподілу визначає щільність тоді, і тільки тоді, коли їхня сумісна щільність задається певною формою.[4]

Теорема Гаммерлі-Кліффорда: розподіл, який задовольняє умову $\pi(x) > 0$ для всіх конфігурацій з $X^{|I|}$ є випадковим полем Маркова тоді і тільки тоді, коли його спільну щільність розподілу можна виразити у наступному вигляді:

$$\pi(x) = \frac{1}{Z} e^{(-\sum_{c \in C} \Phi_c(x_c))} \quad (5)$$

для функція певних функцій $\{\Phi_c\}$,

C – множина усіх клік(кліка – це підмножина вузлів, у якій кожна пара цих вузлів є сусідами один з одним;

Z – множник для нормалізації, який задається наступним чином:

$$Z = \sum_x e^{(-\sum_{c \in C} \Phi_c(x_c))} < \infty \quad (6)$$

Розподіл типу (5) називається розподілом Гіббса.[1]

Означення: Множина випадкових змінних $\{x_i: i \in I\}$ являється випадковим полем Гіббса стосовно системи околів ∂i тоді і тільки тоді, коли конфігурації цієї множини визначається розподілу Гіббса.

Означення: Поле Гіббса називається однорідним, якщо воно не залежить від розташування кліки c в X . Поле Гіббса є ізотропним, якщо воно не залежить від орієнтації c . Якщо випадкове поле Гіббса є ізотропним або однорідним, то тоді значно легше визначити розподіл цього поля. Для того, щоб

спростити нам життя і математичні обрахунки, припускається, що у більшості випадків модель марковських випадкових полів є однорідною.[1]

Ми дізналися про випадкові поля Маркова та випадкові поля Гіббса. В чому ж відмінності між ними? Основна відмінність в тому, що випадкове поле Маркова характеризується локальною властивістю, тим часом як випадкове поле Гіббса характеризується глобальною властивістю(так званий розподіл Гіббса). Теорема Гаммерслі-Кліффорда створює еквівалентність між цими двома властивостями. Відповідно до даної теореми, впливає наступне: $\{x_i: i \in I\}$ являється випадковим полем Маркова над X відносно околу ∂i тоді і тільки тоді, коли $\{x_i: i \in I\}$ є випадковим полем Гіббса над X відносно ∂i . [4]

2.5: Представлення зображень як бінарної моделі

Оскільки ми вже трохи дізналися теоретичних аспектів, час приступити до чогось більш прикладного, а саме до математичного моделювання, яке використовується в цифровій обробці зображень. Припустимо, що у нас чорно-біле зображення, тобто всі пікселі мають лише два кольори: чорний і білий. Тоді можна представити дане зображення, як бінарне, тобто нехай

$$x_i = 1 - \text{піксель чорного кольору};$$

$$x_i = 0 - \text{піксель білого кольору}.$$

Це нам необхідно для того, щоб визначити, що собою являють наші апіорні знання про оригінальне зображення x . [1] Спочатку, потрібно визначити, чи залежить поведінка пікселя x_i від всіх пікселів, які знаходяться біля, тобто від його найближчих сусідів, які визначаються певною моделлю(нагадаємо, що найпопулярніші – чотири, вісім та дванадцять сусідів). Для прикладу, візьмемо схему, в якій модель представлення для пікселя x_i є чотири його сусіди. Припустимо, що ці пікселі забарвлені наступними кольорами і наступним чином:

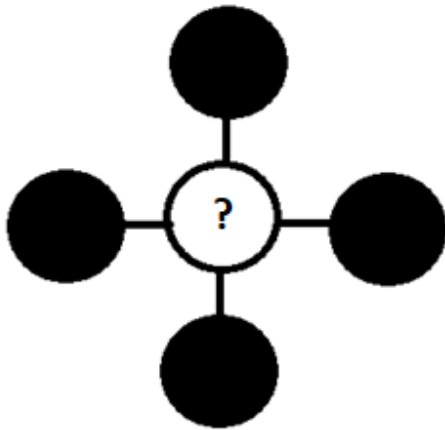


Рис.4

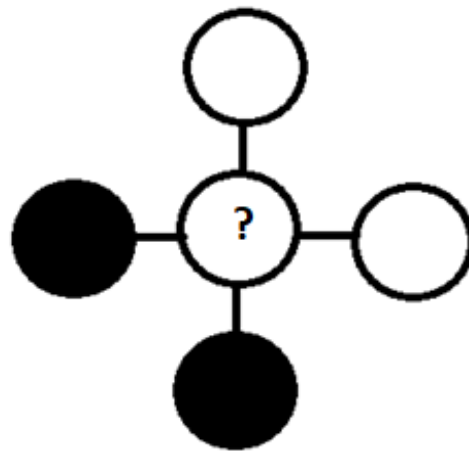


Рис.5

Як можна спостерігати на Рис.4, всі сусідні пікселі зафарбовані в чорний колір. Питання: яка ймовірність, що піксель в середині(x_i) теж чорного кольору? Ця ймовірність буде залежати як від типу розглянутого зображення, так і від його контексту та структури, проте, з точки зору логіки, ця ймовірність повинна збільшуватися разом зі збільшенням кількості сусідніх пікселів чорного кольору.[1] Для вираження такої ймовірності можна використати наступну формулу:

$$\pi(x_i = 1 | \text{кількість сусідніх пікселів чорного кольору}) \propto e^{\beta k} \quad (7)$$

β – якийсь додатний параметр.

Множник для нормалізації має вигляд:

$$\frac{e^{(\beta * \text{кількість сусідніх пікселів білого кольору})}}{e^{(\beta * \text{кількість сусідніх пікселів чорного кольору})}} \quad (8)$$

Як можна побачити з формули (8), у нас немає причин надавати перевагу конкретному виду забарвлення, адже як для $x_i = 1$ так і для $x_i = 0$,

ймовірності будуть однакові, тобто $p = \frac{1}{2}$. При заміні пікселів чорного кольору на пікселі білого кольору, або при заміні пікселів білого кольору на пікселі чорного кольору, конфігурація не зміниться, тобто змінена конфігурація матиме таку ж саму ймовірність, як і початкова конфігурація.[1]

Тепер назріває питання: Яким чином можна задати потенціальні функції, визначені на множині кліків, щоб ці функції мали такі ж локальні характеристики, як і ті, що були розглянуті до цього? Нагадаємо, що таке кліки: кліки – це множини вузлів, задані на системі околу, де пікселі є сусідами попарно. Таким чином, при використанні моделі з чотирьох сусідів, ми матимемо множини з чотирьох елементів, в якій кліками будуть являтися множини найближчих вертикальних пар та найближчих горизонтальних пар. Використовуючи кліку такого типу, можна задати потенціальну функцію:

$$\Phi_C(x_C) = \begin{cases} -\beta, & \text{якщо кліка } C \text{ містить вузли одного кольору} \\ 0, & \text{в іншому випадку} \end{cases} \quad (9)$$

Тому щільність спільного розподілу буде виглядати наступним чином:

$$\pi(x) = \frac{e^{(-\sum_{C \subseteq \mathcal{C}} \Phi_C(x_C))}}{Z(\beta)} = \frac{e^{(\beta \sum_{i \sim j} I_{[x_i = x_j]})}}{Z(\beta)} \quad (10)$$

$i \sim j$ – пари сусідніх пікселів.

Множником для нормалізації є функція від параметра β вигляду:

$$Z(\beta) = \sum_x e^{(\beta \sum_{i \sim j} I_{[x_i = x_j]})} \quad (11)$$

Цей розподіл має ті ж самі повні умовні ймовірності, які були представлені в (7).[1]

Функція спільної щільності з (10) відома як модель Ізінга, названа на честь німецького вченого Ернста Ізінга. Дана модель вперше була представлена в

1925 році. Ізінг змодельював явище феромагнетизму, представивши кожен вузол як спін, який був **орієнтований** вгору(+) або вниз(-).[4] Розглянемо роль моделі Ізінга в цифровій обробці зображень.

2.6: Модель Ізінга

Модель Ізінга являється класичною моделлю в статистичній механіці для пояснення явища феромагнетизму.[1]

В моделі Ізінга розглядається ідеалізована система, в якій частинки, розташовані у вузлах планарної ґратки, взаємодіють між собою. Кожна така частинка має одні з двох магнітних орієнтацій спінів. Такі орієнтації прийнято позначати + або -. Квантова фізика говорить, що спін – це фундаментальна властивість частинки(наприклад, фотона чи ядра). Особливістю даної моделі є те, що частинки перебувають у взаємодії лише зі своїми найближчими сусідами. Загальна енергія системи залежить від внеску кожної частинки, який в свою чергу залежить від орієнтації спіна в порівнянні з іншими сусідами даного спіна. Якщо частинки, які розташовані поруч, мають один і той самий спін, наприклад, $(-1,-1)$ або $(+1,+1)$, то вони перебувають в нижчому стані енергії ніж ті спіни, які мають протилежні спіни $(+1,-1)$ або $(-1,+1)$. Для обрахунку повної енергії системи потрібно знати орієнтацію всіх спінів системи.

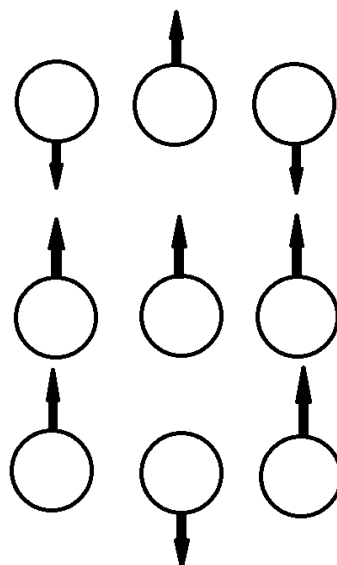


Рис.6. Модель Ізінга

Тепер, нехай змінна x_i означає спін i -ї частинки.

Таким чином, Гамільтоніан моделі Ізінга буде виражатися через наступну формулу:

$$H = - \sum_{i \sim j} J x_i x_j - \sum_j h x_j \quad (12)$$

J – коефіцієнт зв'язку між спінами;

h – коефіцієнт впливу сили зовнішнього поля;

x_i, x_j – спіни у вузлах сітки.

У даній формулі, перший доданок(сума по всіх парах сусідніх вузлів) являється попарними взаємодіями між спінами. Другий доданок(сума по всіх вузлах) являється зовнішньою силою, чий зусилля спрямовані на всіх спінів в одному напрямку.[1]

Таким чином, модель Ізінга має два типи взаємодії:

1. Коефіцієнт h виражає взаємодію зовнішнього поля. Дана величина вказує на те, з якою силою діє поле а також додатню різницю енергії взаємодії одного спіна над іншим. Знак даного коефіцієнта показує, якій орієнтації спіна(вгору або вниз) надано перевагу. Так як кожен спін піддається впливу зовнішнього поля, то для знаходження загального внеску до енергії потрібно просумувати всі вузли.
2. Коефіцієнт J виражає взаємодію сусідніх спінів. Даний коефіцієнт говорить з якою силою пов'язані сусідні спіни а також як сильно вони хочуть розпорядкуватися або впорядкуватися. Знак коефіцієнта J говорить про те, чому хочуть надати перевагу спіни: впорядкуванню чи розпорядкуванню(у фізиці дана поведінка характеризує явища феромагнетизму та антиферомагнетизму). Так як взаємодія виникає

лише між сусідніми спінами, то для знаходження загальног внеску до енергії, потрібно сумувати всі пари вузлів, які є сусідами.[1]

Властивості моделі Ізінга(не всі):[1]

- ❖ Квадратична n -вимірна сітка характеризується тим, що кожен сусід має $2n$ найближчих сусідів;
- ❖ Енергія зв'язку - це енергія взаємодії між двіма суміжними вузлами;
- ❖ Взаємодія в моделі Ізінга відбувається лише з тими вузлами, які суміжні один з одним
- ❖ Усі сусідні вузли розглядаються як незалежні від напрямку, тобто ізотропні

2.7: Алгоритм семплуванні Гіббса

Тепер можна описати алгоритм семплування Гіббса, який належить до класу алгоритмів Монте-Карло:

1. Спочатку обираємо конфігурацію зображення:

$$x = (x_1, x_2, x_3, x_4, \dots, x_{N^2},);$$

2. Проходимося по зображенню послідовно, тобто $1, 2, 3, 4, 5, 6, 7, \dots, N^2, 1, 2, 3, 4, \dots$;
3. Для кожного пікселя n обраховуємо $w_n = \sum_{m \sim n} x_m$, m – сусідні пікселі пікселя n ;
4. Знайдемо наступні ймовірності:

$$p^- = \frac{e^{(-\beta w_n)}}{e^{(-\beta w_n)} + e^{(\beta w_n)}}$$

$$p^+ = \frac{e^{(\beta w_n)}}{e^{(-\beta w_n)} + e^{(\beta w_n)}}$$

5. Генерація нового значення для пікселя $x_n = x_n^{new}$.

$x_n^{new} = 1$ з ймовірністю p^+ та

$x_n^{new} = -1$ з ймовірністю p^- . [1]

2.8: Шум у зображеннях

Моделі, які були розглянуті до цього не враховували наявний шум у зображеннях. Тепер можна і описати модель, яка представляє бінарне зображення і включає не тільки апіорні знання але й відомості про тип шуму у сигналі.

Нехай y_j та y_i є умовно незалежними від x , тобто

$$\pi(y|x) = \pi(x_1|y_1) \dots \pi(y_n|x_n)$$

Також можна визначити, яким чином бінарне випадкове поле Маркова може бути представлено як апіорне знання при позбавленні зображень від шуму.

Як приклад, розглянемо чорно-біле зображення з наявним у ньому шумом, наприклад, шумом «сіль-перець» .[3]

2.9: Тип шуму «Сіль-перець»

Даний вид шуму виникає тоді, коли ймовірність правильного значення пікселя, тобто колір, записаний у вигляді бінарної моделі є неправильною. Цей вид шуму підпорядковується закону Бернуллі.

За припущенням, ймовірність помилки p є константною, крім того, значення кожного окремого пікселя записуються незалежно один від одного.

$$\pi(y_i|x_i) = \begin{cases} 1 - p, & y_i = x_i \\ p, & y_i \neq x_i \end{cases}$$

$p = \frac{1}{2}$ означає, в у відсутня інформація про x . Проте апостеріорний розподіл може мати наступний вигляд:

$$h_i(x_i, y_i) = I_{[x_i=y_i]} \log\left(\frac{1-p}{p}\right)$$

$I_{[x_i=y_i]}$ – функція індикатор.[3]

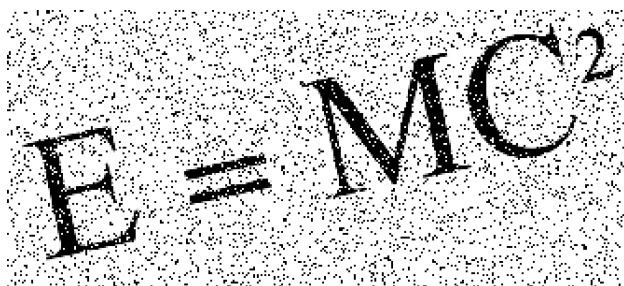
РОЗДІЛ 3: Реалізація алгоритму обробки зображень

Реалізація алгоритму семплування Гіббса написана з використання мови програмування Python, а зокрема модулів `math`, `numpy` та `matplotlib`. Даний алгоритм працює з чорно-білими зашумленими зображеннями. Як працює? На вхід отримується зображення з шумом, потім воно перетворюється в модель Ізінга і обраховує ймовірності для нового зображення. Потім нова модель Ізінга перетворюється у нове чорно-біле зображення і зберігається. Функції:

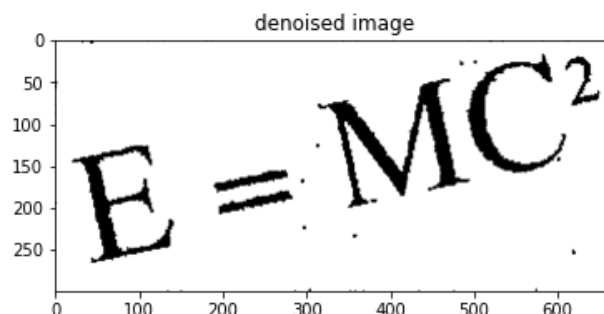
- `read_image` - зчитує зображення за шляхом, перетворює в модель Ізінга додає до цього зображення обідок з нулів і повертає нове зображення;
- `return_image` – перетворює модель Ізінга в чорно-біле зображення і повертає його;
- `save_image` – зберігає зображення за допомогою засобів бібліотеки `matplotlib`;
- `count_proba` – обраховує ймовірності для кожного пікселя по відношенню до його сусідів і повертає 1 або -1;
- `sampling` – будує матрицю зображення на основі ймовірностей, обрахованих за допомогою функції `count_proba`;
- `main` – точка входу в програму.

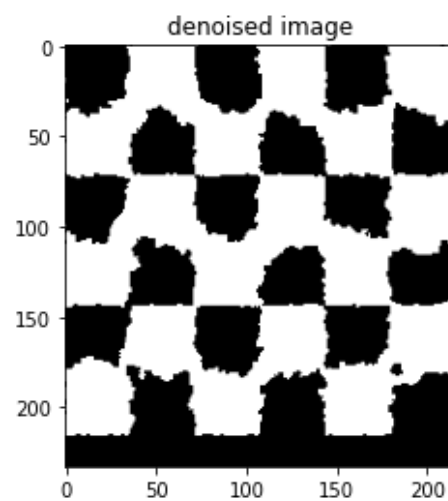
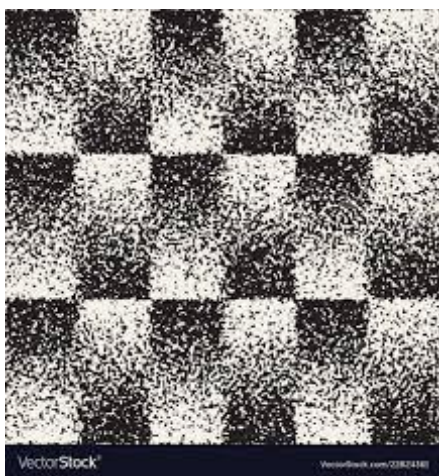
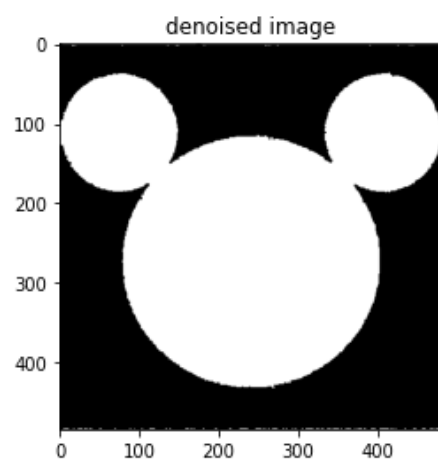
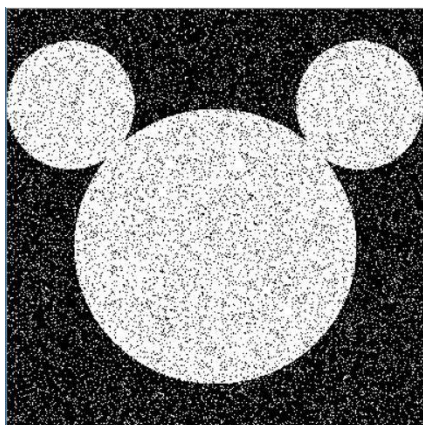
Результати роботи алгоритму:

Зашумлене



Відновлене





Як можна помітити вище, відновлення зашумлених зображень не є досконалим. Якщо перші два зображення були досить простими, то зображення з шаховою дошкою було відновлено не ідеально. Крім того, складність алгоритму буде $O(n^2)$, тому час виконання буде великими. У моєму середовищі виконання час обробки одного зображення складав близько трьох хвилин, що з точки зору програмування є дуже багато. Код для даного алгоритму буде представлено в додатках.

Висновки

В даній роботі було розібрано баєсівський підхід в обробці зображень та показано роботу алгоритму семплування Гіббса на основі моделі Ізінга. В подальшому планується розглянути інші алгоритми для дешумізації зображень, які використовують модель Ізінга, розробити модель для моделювання кольорових зображень а також покращити на програмному рівні швидкість роботи алгоритму семплування Гіббса.

Додаток

```
import math
import numpy as np
import matplotlib.pyplot as plt
from google.colab import drive

def count_poba(i, j, Y, X):
    markov_field = [Y[i - 1, j], Y[i, j - 1], Y[i, j + 1], Y[i + 1, j], X[i, j]]
    w = alfa * markov_field[-1] + beta * sum(markov_field[:4])
    probability = 1 / (1 + math.exp(-2*w))
    return (np.random.rand() < probability) * 2 - 1

def read_image(file):
    img = plt.imread(file)
    gray_image = np.dot(img[:, :, :3], [0.2989, 0.5870, 0.1140])
    gray_image = np.where(gray_image > 0.5, 1, -1)
    padded_image = np.zeros([gray_image.shape[0] + 2, gray_image.shape[1] + 2])
    padded_image[1:-1, 1:-1] = gray_image
    return padded_image

def return_image(filename, amount_of_steps, amount_of_samples):
    posterior = sampling(filename, amount_of_steps, amount_of_samples)
    denoised = np.zeros(posterior.shape, dtype=np.float64)
    denoised[posterior > 0.5] = 1
    return denoised[1:-1, 1:-1]

def sampling(filename, amount_of_steps, amount_of_samples):
    X = read_image(filename)
    posterior_img = np.zeros(X.shape)
    Y = np.random.choice([1, -1], size=X.shape)
    for step in range(amount_of_steps + amount_of_samples):
        for i in range(1, Y.shape[0]-1):
            for j in range(1, Y.shape[1]-1):
                y = count_poba(i, j, Y, X)
                Y[i, j] = y
                if y == 1 and step >= amount_of_steps:
                    posterior_img[i, j] += 1
    posterior_img = posterior_img / amount_of_samples
    return posterior_img

def save_image(denoised_image):
    plt.imshow(denoised_image, cmap='gray')
    plt.title("denoised image")
    plt.savefig('/content/drive/MyDrive/DataAnalysis/denoise_image5.png', bbox_inches='tight')
    plt.close()

if __name__ == '__main__':
```

```
alfa = 1
beta = 1
total_samples = 1000
burn_in_steps = 100

drive.mount("/content/drive", force_remount=True)
denoised_img = return_image("/content/drive/MyDrive/DataAnalysis/4.png", amount_of_steps=burn_in_steps, amount_of_samples=total_samples)
save_image(denoised_img)
```

Література

- [1] Герхард Винклер. Анализ изображений,случайные поля и методы Монте-Карло на цепях Маркова. Матиматические основы. Второе издание. Новосибирск, академическое издательство «ГЕО», 2009.
- [2] Charles A. Bouman. Markov Random Fields and Stochastic Image Models. School of Electrical and Computer Engineering, Purdue University. Washington, D.C. , 1995.
- [3] Alex F Bielajew. Fundamentals of the Monte Carlo method for neutral and charged particle transport. The University of Michigan Department of Nuclear Engineering and Radiological Sciences. The University of Michigan, 2001
- [4] Mohammed Sheikh. Markov Chain Monte Carlo and Image Restoration.December 19, 2012.
- [5] Merrilee A. Hurn, Oddvar K. Husby, Havard Rue. A Tutorial on Image Analysis. January, 2003