ЗМІСТ

[ВСТУП 2](#_Toc514520522)

[РОЗДІЛ 1. ТЕОРЕТИЧНЕ ПІДГРУНТЯ КОМП’ЮТЕРНОГО ЗОРУ 4](#_Toc514520523)

[1.1 Ідея розпізнавання тексту. 4](#_Toc514520524)

[1.2 Усунення шуму. 5](#_Toc514520525)

[1.3. Бінаризація зображення. 6](#_Toc514520526)

[1.4. Вирівнювання зображення. 7](#_Toc514520527)

[1.5. Сегментація символів. 9](#_Toc514520528)

[1.5. Виявлення ліній та слів 14](#_Toc514520529)

[1.5.1. Перетворення Хафа 14](#_Toc514520530)

[1.5.2 Перетворення Фур’є 15](#_Toc514520531)

[1.5.3 Аналіз гістограми вертикальної проекції. 16](#_Toc514520532)

[1.6. Класифікація символів 17](#_Toc514520533)

[1.6.1. Мотивація нейронної мережі. 17](#_Toc514520534)

[1.6.2. Штучний нейрон. 18](#_Toc514520535)

[1.6.3. Функції активації 18](#_Toc514520536)

[1.6.4. Структура нейронної мережі. 20](#_Toc514520537)

# ВСТУП

Оптичне розпізнавання символів – це механічне або електронне перетворення зображень друкованого або рукописного тексту у електронний текст. Це може бути сканований документ, сфотографований текст або пейзажна фотографія (текст на дорожніх знаках, рекламі тощо). Оптичне розпізнавання символів набуло великої популярності за часів НТР.

Початком розпізнавання фізичного тексту можна вважати 1914 рік, коли були реалізовані перші для телеграфії та створення приладів читання тексту для людей з вадами зору. У 1914 році Емануель Гольдберг розробив прилад, який зчитував символи і перетворював їх в стандартний код для телеграфа. В той же час ірландський фізик Едмунд Фурньє розробив прилад, який сканував текст і перетворював символи у звукові сигнали (прилад отримав назву «оптофон»).

До появи сучасних комп’ютерів практично уся інформація зберігалась на паперових носіях (книги, наукові статті, журнали, історичні документі тощо). З появою електронних обчислювальних машин з’явилась можливість зберігати інформацію в електронному вигляді, що має цілий ряд переваг:

* Зручний і безпечний спосіб зберігання
* Можливість зробити резервну копію
* Можливість опрацювання інформації комп’ютером
* Можливість одночасного спільного користування

Зважаючи на цей неповний список переваг, більшість нової інформації з’являється спершу в електронному вигляді, а уже потім при потребі з’являються друковані примірники. Постає проблема: перетворити існуючі друковані документи, у електронні примірники. Проблема розповсюджується і на сучасні документи, які не мають вмонтованих електронних носіїв, розпізнавання номерних знаків механічних транспортних засобів тощо.

Кардинально інше застосування оптичного розпізнавання символів – допомога людям з вадами зору (читання книг, орієнтація в просторі тощо).

Процес переведення інформації в цифрову форму (дискретні сигнальні імпульси) називається оцифруванням.

Складність задачі визначається відсутністю чіткого математичного означення «літера», «слово» ті інших термінів (в сенсі «коли літера перестає бути літерою?»), великою різноманітністю шрифтів, розмірів букв, кольорів, стилів тексту тощо.

# РОЗДІЛ 1. ТЕОРЕТИЧНЕ ПІДГРУНТЯ КОМП’ЮТЕРНОГО ЗОРУ

## 1.1 Ідея розпізнавання тексту.

Растрове чорно-біле комп’ютерне зображення представляє собою матрицю чисел, де кожен елемент матриці відповідає за інтенсивність випромінюваного світла в конкретній точці зображення. Кольорове складається з елементів, які містять значення по кожному кольоровому каналу.

Глибина кольору – це кількість бітів, відведених для зберігання кольору.

Кольоровий простір – це математична модель кольору (наприклад, RGB – суміш червоного, зеленого та червоного кольорів).

В загальному процес розпізнавання тексту включає в собі послідовність наступних етапів

* Вирівнювання зображення – якщо зображення тексту не є паралельним і OX, зображення необхідно повернути на певний кут
* Усунення шуму – процес усунення небажаних похибок сигналу
* Бінаризація – процес перетворення зображення до зображення з глибиною кольору в 1 біт.
* Виявлення ліній та слів – отримання даних геометричного розташування рядків та слів
* Сегментація символів – отримання даних геометричного положення символу
* Класифікація символів – визначення семантики символу

Послідовність цих етапів не є строгою, тому можна процес перетворення реорганізувати наступним чином

1. Усунення шуму
2. Бінаризація зображення
3. Вирівнювання зображення
4. Сегментація символів
5. Виявлення ліній та слів
6. Класифікація символів
7. Аналіз добутої інформації

Наведений вище порядок має кращу швидкодію в порівнянні з традиційним шляхом.

Будемо вважати, що вхідні данні – 24 бітне RGB зображення.

Позначимо це зображення як

,

де вектор – це елемент матриці I, який знаходиться в рядку та стовпчику. Елементи матриці зображення називатимемо пікселями. Кожен піксель кольорового зображення; інтенсивність червоного, зеленого та синього кольорів.

## 1.2 Усунення шуму.

Для усунення шуму зручно використати фільтр Гауса для розмиття. Для цього застосуємо операцію згортки зображення.

Згортка зображення з ядром та якорем являє собою заміну кожного пікселя вихідної матриці на суму добутків відповідних пікселів вхідною матриці на коефіцієнти ядра , при умові що положення елемента ядра збігається з положенням вихідного пікселя .

Операція згортки позначається

Враховуючи дискретність зображення інтегральну форму згортки можна записати в простішому вигляді

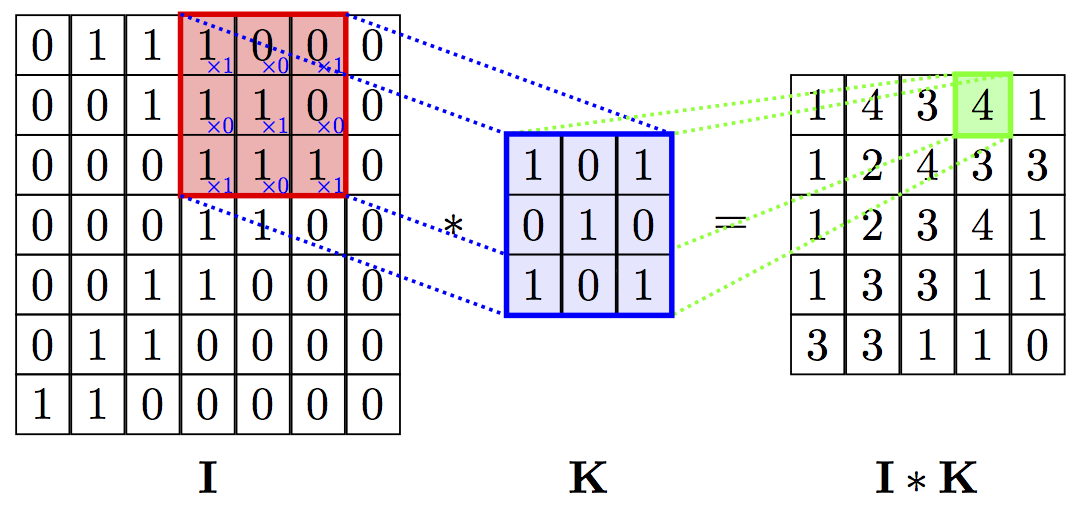


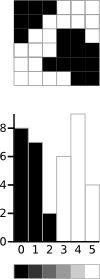
Рис. 1

Для застосування фільтра Гауса достатньо вибрати ядро згортки з Гаусівським розподілом

## 1.3. Бінаризація зображення.

Бінаризація зображення – конвертація зображення у одноканальне зображення з глибиною кольору в 1 біт.

Для біниразації необхідно використати порогові операції. Вибравши поріг T, застосуємо формулу:



Проблема полягає у виборі порогового значення . Для вибору цього значення використаємо алгоритм Отсу.

Алгоритм однозначно шукає таке значення параметру , який мінімізує покласову варіансу, визначену як зважена варіанс двох класів.

Рис. 2

(t) – ймовірність класів, (t) – варіанси класів.

## 1.4. Вирівнювання зображення.

Вирівнювання зображення – процес пошуку такого кута повороту, при якому лінії тексту будуть точно горизонтальними. Для вирівнювання тексту використаємо метод гістограм. Розглянемо бінаризоване зображення ,

На основі цього зображення побудуємо вертикальну гістограму такої ж висоти, як і саме зображення. Кожен -й елемент вектора-гістограми буде вказувати на кількість нульових елементів матриці в -тому рядочку.

Рядки тексту розділені між собою вертикальним відступом. Відступ на гістограмі буде відображено малою величиною, в той час як рядок текст – значною кількістю ненульових елементів.



Найкращий поворот тексту, який міститиме усі горизонтальні рядки тексту буде відображений зубчастою гістограмою – локальні максимуми будуть чергуватися з локальним мінімумом. Проте, пошук локальних максимумів та аналіз їх розташування не дасть точного результату роботи алгоритму з огляду на похибки. Наприклад, кожна область зростання гістограми міститиме декілька локальних мінімумів.

Різкі спадання та зростання функції можна охарактеризувати девіацією розподілу.

Рис. 3

Таким чином, необхідно визначити такий кут повороту , що зображення, повернуте на цей кут продукує гістограму з максимальною девіацією.

Таблиця 1

|  |  |
| --- | --- |
|  | C:\Users\Admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\inithist.png |
| C:\Users\Admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\bettertext.png | C:\Users\Admin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\better.png |
|  |  |

## 1.5. Сегментація символів.

Бінаризація зображення допомагає відділити фон від об’єктів, які нас цікавлять. Використавши бінаризацію методом Отсу, отримаємо літери чорного кольору, а фон – білого кольору. Літери друкованого тексту відділені одна від одної і не мають точок дотику. Як наслідок, на обробленому зображені кожна літера утворюватиме окрему однозв’язну область. Для виокремлення кожної окремої літери необхідно отримати множину однозв’язних областей. Для кожної такої множини необхідно знайти ламану, яка описує її контур. Виокремивши контур області, можна отримати найменшу прямокутну область, яка містить в собі літеру (сторони прямокутника паралельні осям координат). Зробити це можна знайшовши у контурі чотири точки з мінімальними та максимальними координатами по кожній з осей координат. Прямокутник буде описуватись такими точками: , та ,. В загальному для ефективного пошуку контурів областей використовують практичні наслідки теореми Гріна та моменти зображення. Проте для бінарного зображення у випадку пошуку контуру літери зручно скористатися алгоритмом пошуку кутів Кенні. Алгоритм був розроблений професором каліфорнійського університету у 1986 році.

Рис. 4

Рис. 5



Алгоритм Кенні:

1. Застосування згортки з ядром Гаусівського фільтру з метою усунення шумів
2. Пошук градієнтів зображення

Градієнт одновимірної функції визначається як похідна функції

і визначає швидкість зростання функції у напрямку осі . Оскільки вхідне зображення можна розглядати як дискретну функцію , то вектор-градієнт визначається як вектор часткових похідних

З огляду на дискретність зображення, найменший можливий приріст аргументу функції .

Таким чином похідна зображення – різниця значень сусідніх пікселів – елементів матриці. Для практичних цілей необхідно володіти значеннями похідної в усіх точках, тобто обчислити дві матриці та . Ці матриці зручно обчислювати операцією згортки з ядрами

Наприклад,

Для задачі пошуку градієнта математичне означення похідної підходить ідеально і дає максимально можливу точність вектор-градієнта в кожній точці. Проте, реальні зображення містять багато похибок (наприклад похибка інтерполяції фотонів під час створення зображення камерою), тому є сенс враховувати не лише точно два сусідні пікселі по чи , а деякі сусідні пікселі в певному околі точки. Для цих цілей зручно застосовувати оператори Прюітта чи Соболя. Наприклад, застосуємо такі ядра згортки оператора Соболя для знаходження градієнту зображення, проілюстрованого на рис. 4:

Рис. 6

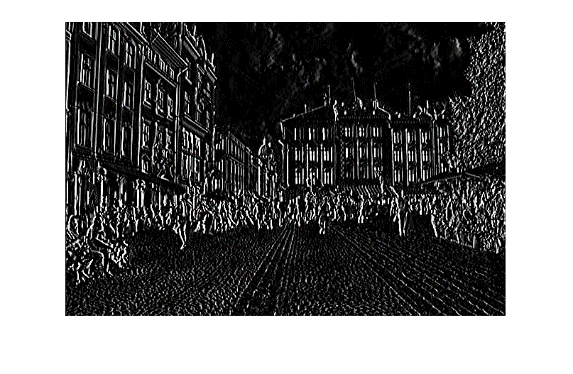


Рис. 7 Згортка ядром М

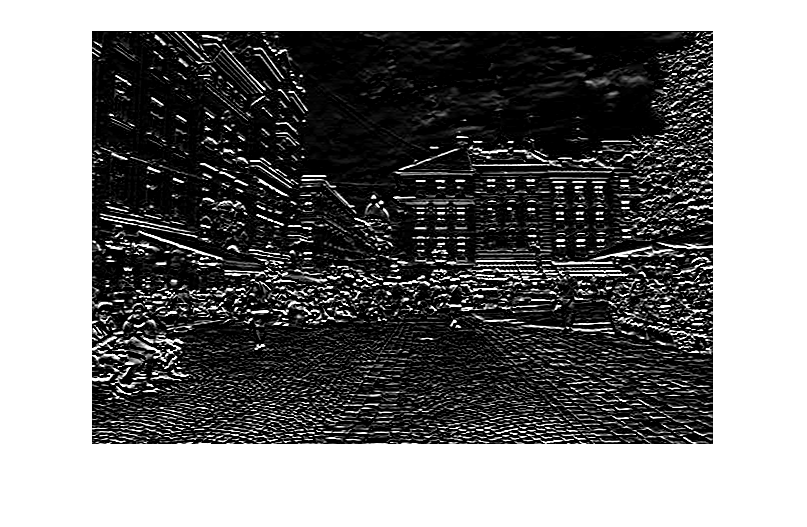


Рис. 8 Згортка ядром H

Швидкість зміни градієнту можна обчислити як

Кут нахилу обчислюємо за формулою оберненого тангенсу двох змінних:

Ця функція шукає кут нахилу, враховуючи напрямок вектора . Знайдений кут округлюємо до одного з чотирьох напрямків . Кути нахилу кожного з кутів округлюємо до тих же значень відповідно.

1. Застосування немаксимального усунення (non-maximum suppression) з метою усунення хибних кутів

Проблема оператора Соболя для визначення кутів у тому, що результатом роботи не є тонка лінія.

Рис. 11

Рис. 99



Рис. 1010



Для знаходження найкращої точки використовується метод немаксимального усунення. У напрямку вектор-градієнта знаходимо локальний максимум. Усі інші точки прирівнюємо до нуля.

Рис. 13

Рис. 112





1. Застосування технології подвійного порогового значення

Для видалення незначних змін градієнту, виберемо два порогові значення

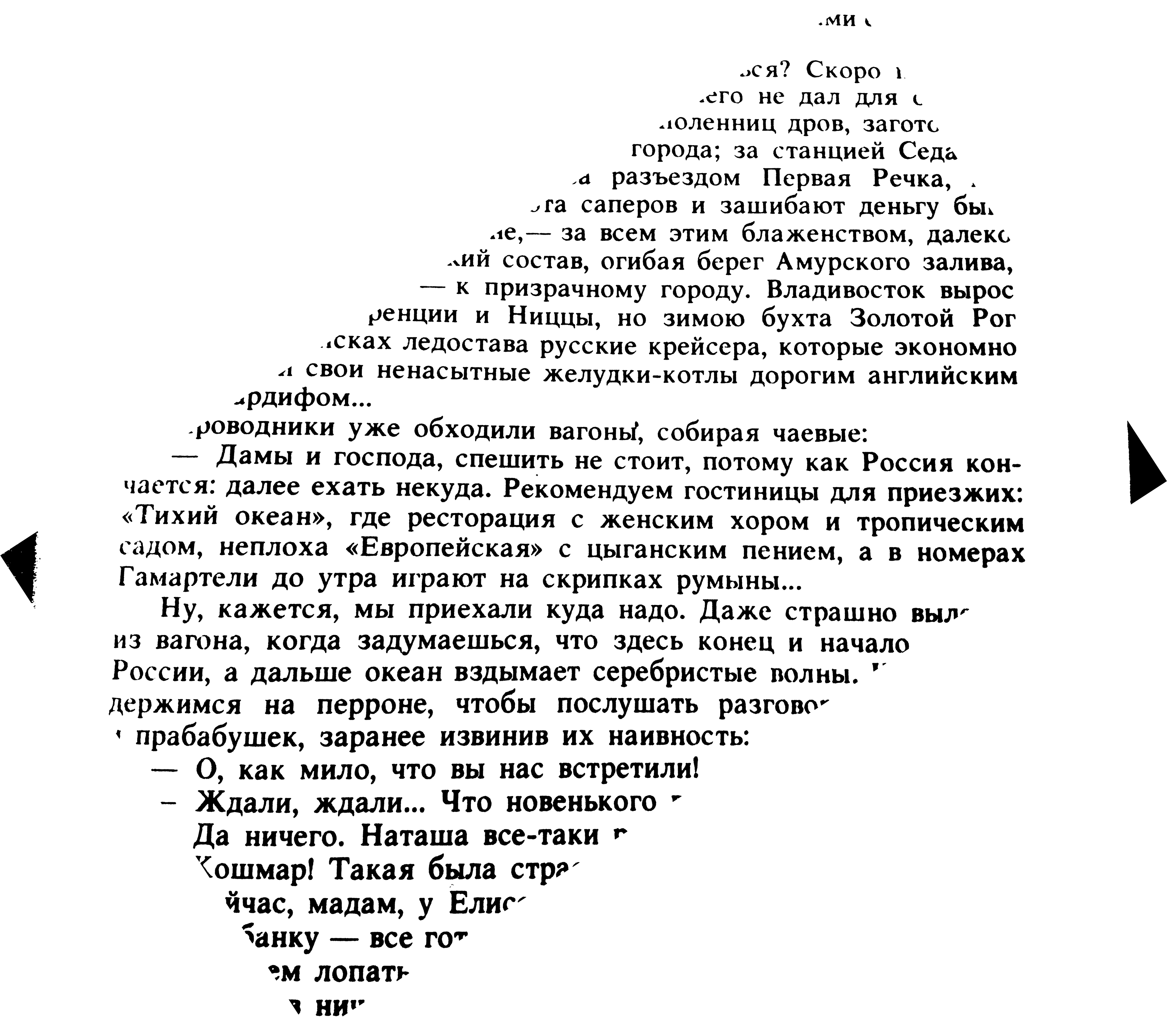
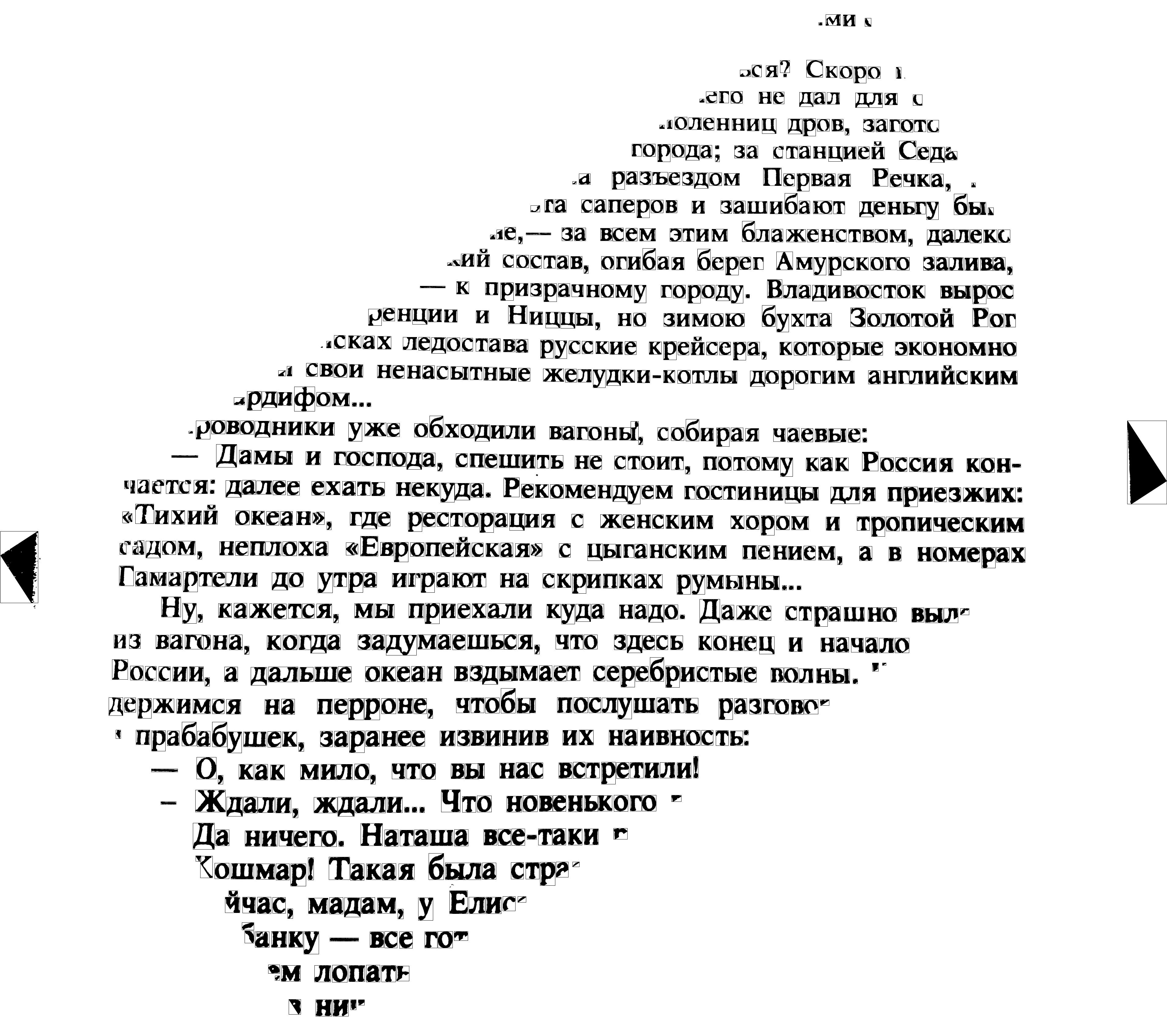
Утворимо нову матрицю за формулою:

Рис. 12

Рис. 13



Винятковими літерами української мови є «і», «ї», «Ї», «й», «Й», оскільки вони не утворюють однозв’язну область.



Усі виняткові букви мають 1-2 невеликі додаткові області зверху. Для того, щоб з’єднати ці області зручно використати морфологічні операції. Ці операції використовують бінарне ненульове ядро з якорем . Якщо , то відповідний елемент не враховується під час обчислень

Операція розширення (dilation) – вихідне зображення формується за правилом



Рис. 14

Операція розмиття (erosion) – вихідне зображення формується за правилом



Рис. 15

На основі комбінацій цих двох базових операцій створюються багато інших – відкриття, закриття, морфологічний градієнт тощо.

Для з’єднання двох областей зручно використовувати відкриття – застосування операції розмиття і розширення

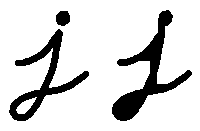


Рис. 16

Якщо задати вертикальне ядро

То операцією відкриття можна з’єднати вертикальні області

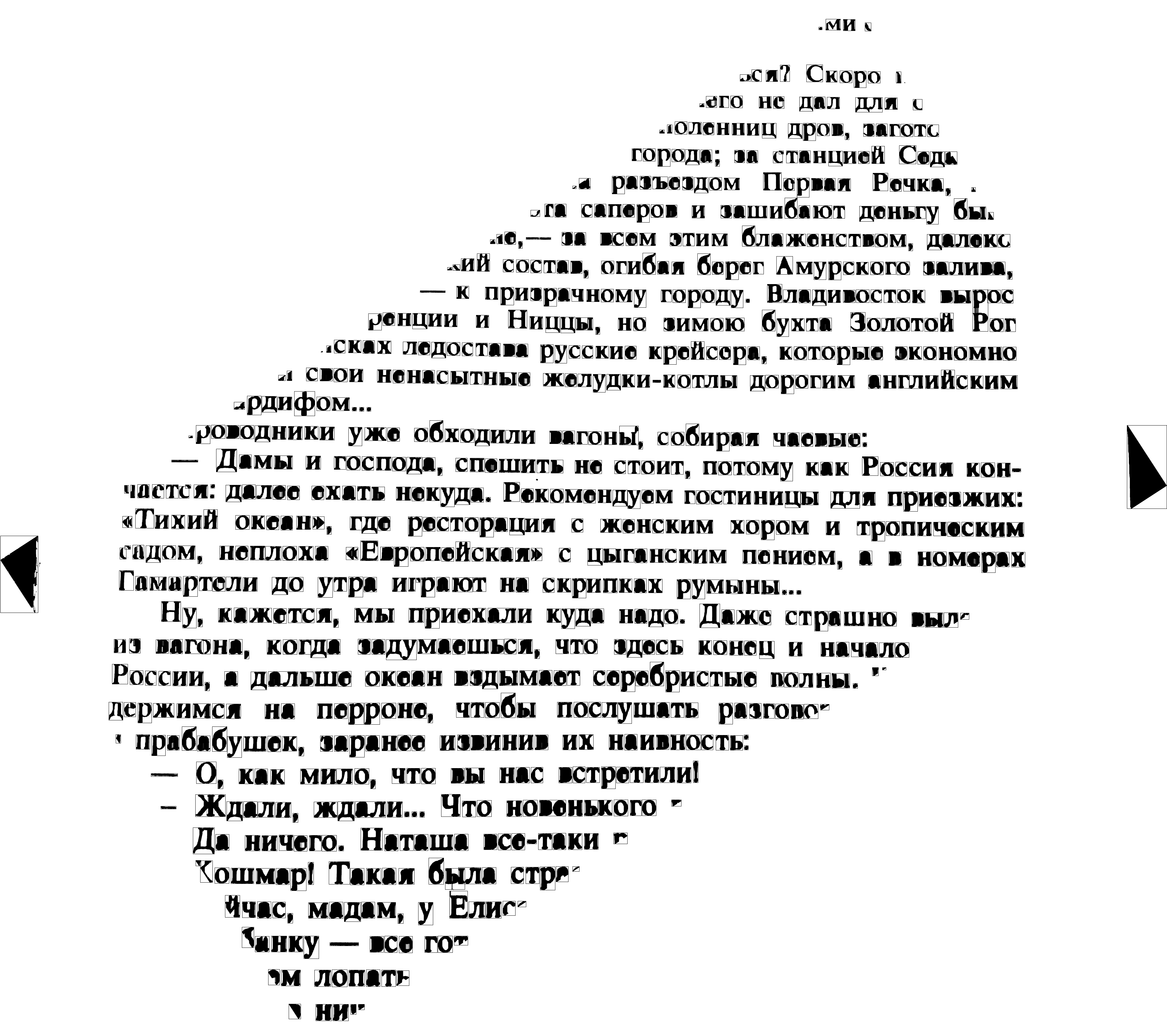
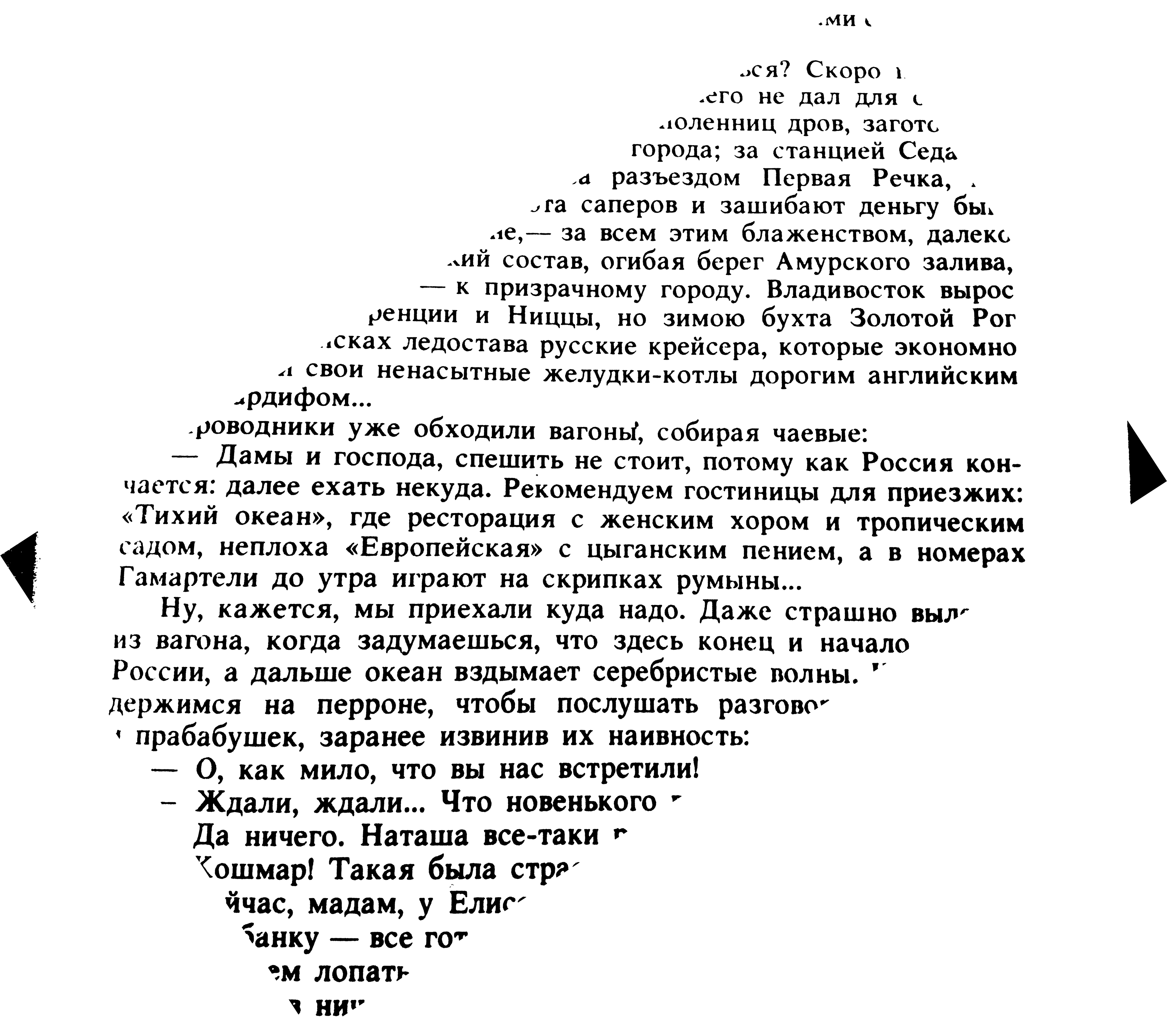
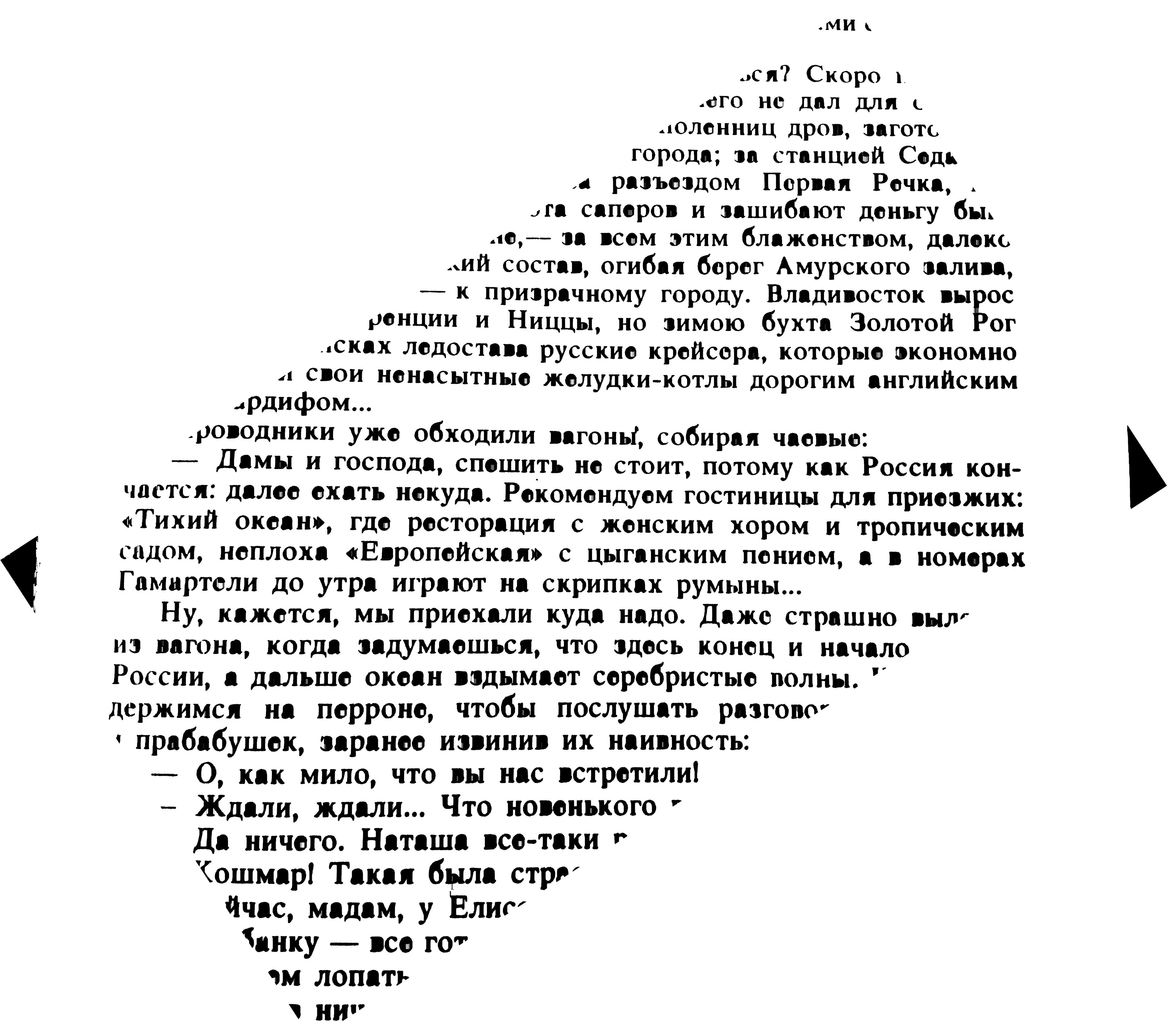


Рис. 17

Отримані області можна накласти на оригінальне зображення зі зміщенням в половину висоти ядра (морфологія з несиметричним ядром зміщує зображення) і отримати оригінальні букви.

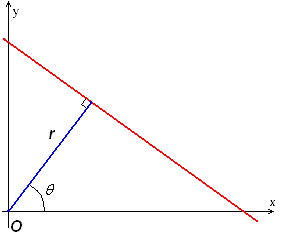
1.5. Виявлення ліній та слів

### 1.5.1. Перетворення Хафа

Зазвичай, для пошуку параметричних кривих використовується перетворення Хафа. Найпростіший випадок – пошук прямих . Пряма розглядається в такому вигляді

– пряма. Такий простір називається простором Хафа. Точка простору Хафа відповідає прямій Евклідового простору. Реалізація алгоритму проста. Розбиваємо простір Хафа скінченними елементами однакового розміру (зазвичай квадратами) і кожному елементу ставимо у відповідність початкове число – 0. На бінарному зображенні для кожної ненульової точки будують

Рис. 18



з певним кроком . Перетворюємо пряму у вигляд . Знаходимо скінченний елемент простору Хафа, котрому належить точка та інкрементуємо відповідне значення. Візуалізація кількості голосів кожного елементу виглядає так:

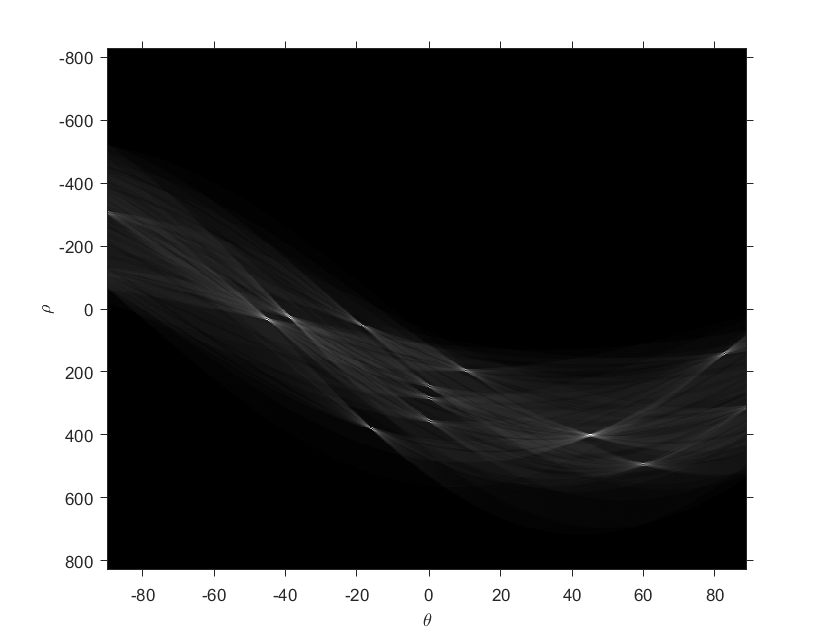


Рис. 19 Лінії у просторі Хафа

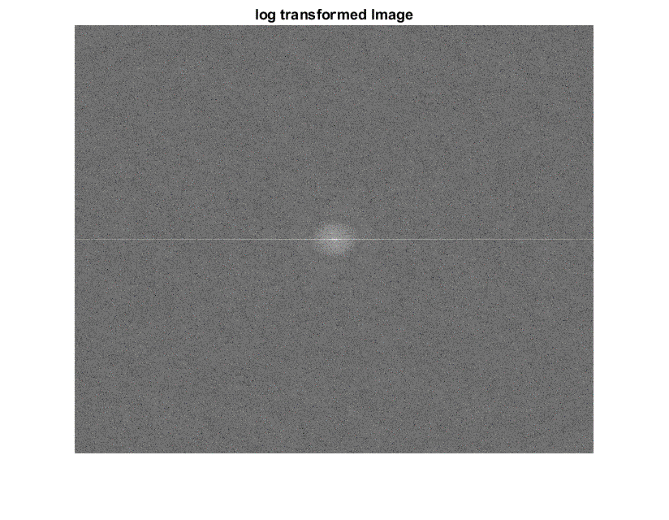
Ті точки, кількість голосів яких є найбільшою, репрезентують лінію у Евклідовому просторі.

### 1.5.2 Перетворення Фур’є

Паралельні лінії тексту утворюють чергування чорних і білих ліній на зображенні. Для пошуку частотних областей зазвичай використовують двовимірне перетворенню Фур’є.

Враховуючи дискретність зображення, можемо переписати формулу у дискретному варіанті – двовимірне дискретне перетворення Фур’є.

Головна ідея перетворення Фур’є полягає в розбитті сигналу зображення на компоненти синуса та косинуса (, формула Ейлера). Синус та косинус – тригонометричні функції з періодом, що дозволяє ефективну шукати частотні області сигналу. Для візуалізації результату перетворення використаємо логарифмічну шкалу



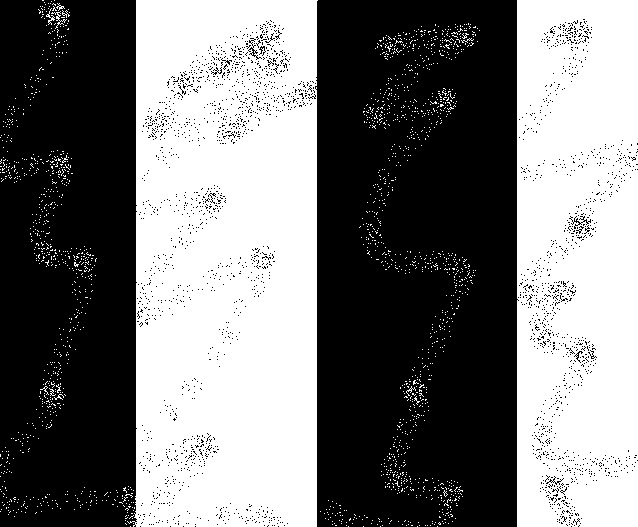
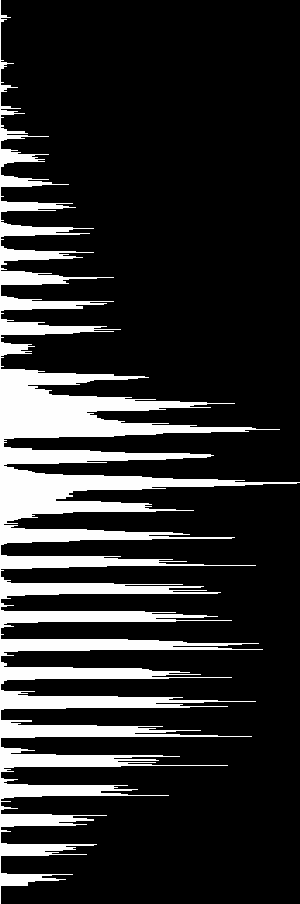


Рис. 20 Магнітуда частотних областей

Лінії тексту чергуються з білими міжрядковими інтервалами. Це спричиняє залежність кількості нульових від наявності лінії в конкретному рядку матриці. Порахуємо гістограму вертикальної проекції зображення.

### 1.5.3 Аналіз гістограми вертикальної проекції.



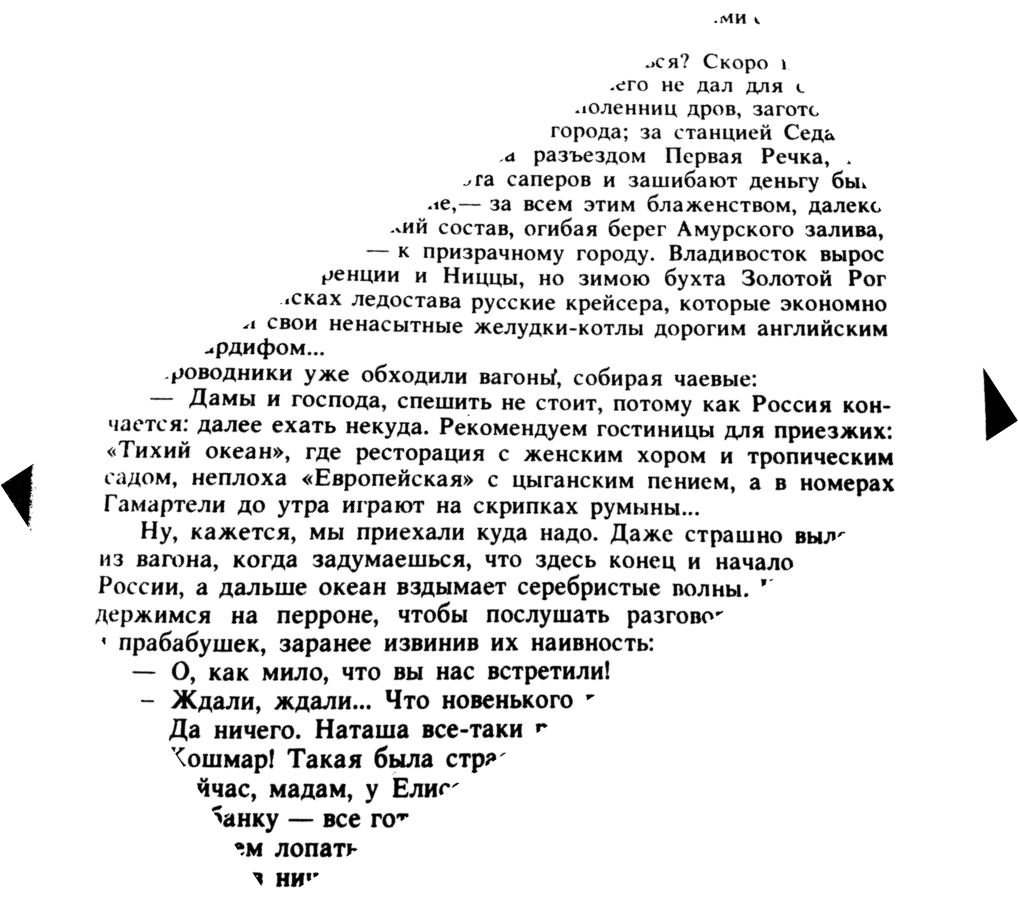
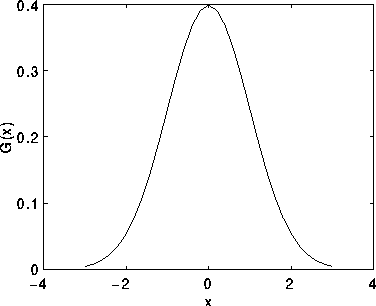


Рис. 21 Гістограма вертикальної проекції тексту

Таким чином кожен рядок тексту продукує локальний максимум гістограми вертикальної проекції зображення. Проблема полягає в знаходженні максимумів, оскільки кожен рядок тексту містить декілька таких точок. Використання ідеї немаксимального усунення (non-maximum suppression) дає погану точність результату.

Застосуємо ідею Гаусівського фільтру. Побудуємо функцію одновимірного Гаусівського розподілу.



Застосуємо операцію згортки до вектор-гістограми

Рис. 22 Розподіл Гауса

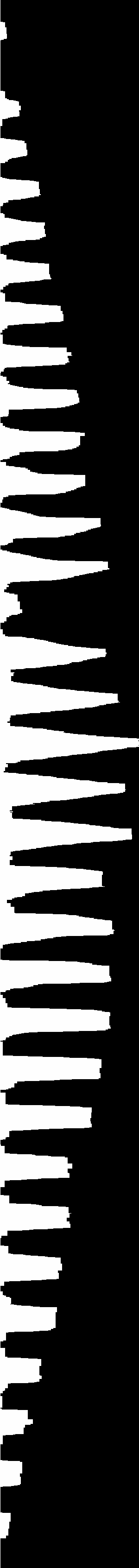
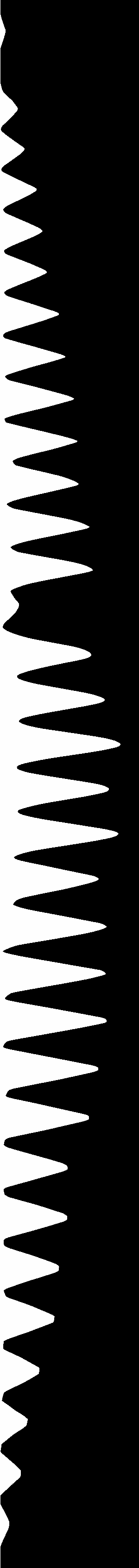


Рис. 23 Фільтрування гістограми вертикальної проекції тексту

Таким чином задача пошуку ліній зводиться до задачі знаходження локальних максимумів гістограми:

Знайдені точки будуть відповідати середині висоти кожного рядка.

## 1.6. Класифікація символів

### 1.6.1. Мотивація нейронної мережі.

Кожен символ, який необхідно класифікувати – це матриця чисел, яка є частиною матриці зображення. Завдання полягає у розробці алгоритму, який на вхід отримує зображення символу, а результатом роботи якого – клас символу. Для класифікації використаємо нейронну мережу. В українській мові є 33 літери – 33 класи символів. Кожні символи з одного класу під час друку на фізичному носії інформації виглядають однаково. Проте, під час фотографії ці дані спотворюються (шум зображення, афінні та перспективні перетворення тощо). Складність завдання в тому, що немає чіткого математичного означення символу, а тим більше його належність до якогось класу. Оскільки, кожен символ, який подається на вхід – матриця чисел, шуканий алгоритм буде являти собою функцію . Причому,

Функція – випадкове спотворення символа . Символ представлений у вигляді вектора розмірності та складається з рядків відповідної матриці зображення. Вигляд функції задаватиметься структурою нейронної мережі та взаємозв’язків нейронів.

### 1.6.2. Штучний нейрон.

Штучний нейрон має такий загальний вигляд

Елементи – вхідні дані нейрона, , – функція активації. Для зручнішого практичного використання візьмемо . Тоді загальний вигляд нейрону матиме такий вигляд

Коефіцієнт зміщення дозволяє зміщати результат функції активації по осі ОХ, що в багатьох випадках є визначальним фактором під час тренування нейронної мережі.

### 1.6.3. Функції активації

Розглянемо деякі функції активації:

Таблиця 2

|  |  |
| --- | --- |
| Крокова функція |  |
| Тотожне відображення |  |
| Сигмоїд |  |
| Гіперболічний тангенс |  |
| Випрямляюча функція активації |  |
| Обернений тангенс |  |
| Гладкий сигнум |  |

Немає чітких рекомендацій до застосування конкретної функції активації. Кожна функція може давати кращий та гірший результат точності апроксимації та швидкості збігання функції.

### 1.6.4. Структура нейронної мережі.

Множина нейронів утворює структуру нейронної мережі. Сама структура поділяється на шари нейронів. Лівий шар нейронів, який отримує вхідні дані називається вхідним шаром, правий – вихідний шар; усі інші шари – приховані шари.

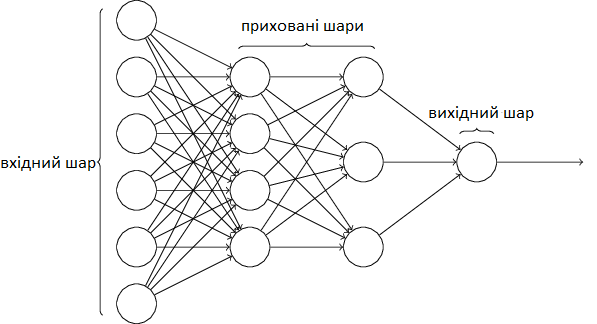


Рис. 24 Структура нейронної мережі

Оскільки кількість нейронів у кожному шарі є сталою для конкретної моделі, тому й кількість вхідних даних також має бути сталою. Для завдання класифікації символів має бути сталою розмірність вектора.

Процес тренування нейронної мережі зводиться до задачі оптимізації, де необхідно знайти такі коефіцієнти , що результат роботи нейронної мережі буде мати найменше відхилення від реального результату на наборі даних для тренування. За умови правильних даних для тренування, нейронна мережа узагальнить кожен клас та видаватиме коректний результат на більшості вхідних даних.