



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

## РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

*К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ*

*НА ТЕМУ:*

«Классификация методов борьбы с шумами в  
изображениях»

Студент ИУ7-75Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

А. Н. Прянишников  
(И.О.Фамилия)

Руководитель

\_\_\_\_\_  
(Подпись, дата)

В. П. Степанов  
(И.О.Фамилия)

2022 г.

# РЕФЕРАТ

Расчетно-пояснительная записка 25 с., 11 рис., 1 табл., 25 ист.

В работе представлена классификация существующих методов борьбы с шумами в изображениях.

**Ключевые слова:** шум, фильтрация, алгоритм, нейронная сеть, фотография, изображение.

Проведен анализ предметной области: определены источники шума и классификация шумов. Представлен общий алгоритм работы всех методов устранения шумов. Описаны существующие способы обнаружения и удаления шумов из изображений. Определены критерии сравнения алгоритмов. Приведена классификация существующих алгоритмов.

# СОДЕРЖАНИЕ

<b>РЕФЕРАТ</b>	<b>3</b>
<b>ВВЕДЕНИЕ</b>	<b>5</b>
<b>1 Анализ предметной области</b>	<b>6</b>
1.1 Причины появления шумов в изображениях . . . . .	6
1.2 Классификация шумов . . . . .	6
1.2.1 Гауссов шум . . . . .	7
1.2.2 Шум соли и перца . . . . .	7
1.2.3 Спекл-шум . . . . .	8
<b>2 Обзор существующих методов</b>	<b>9</b>
2.1 Общий алгоритм работы фильтров . . . . .	9
2.2 Медианный фильтр . . . . .	10
2.3 Гауссовский фильтр . . . . .	12
2.4 Билатеральный фильтр . . . . .	13
2.5 Алгоритм Цзяньвэй . . . . .	14
2.6 DnCNN . . . . .	16
2.7 RIDNet . . . . .	17
<b>3 Критерии сравнения алгоритмов</b>	<b>19</b>
3.1 PSNR . . . . .	19
3.2 SSIM . . . . .	19
<b>4 Классификация существующих решений</b>	<b>21</b>
<b>ЗАКЛЮЧЕНИЕ</b>	<b>22</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>23</b>
<b>ПРИЛОЖЕНИЕ А</b>	<b>25</b>

# ВВЕДЕНИЕ

Цифровой шум в изображениях появляется вследствие свойств фотона света или вмешательства человека [1]. Он оказывает огромное влияние на восприятие человеком качества картинки, а также ухудшает результаты работы алгоритмов, анализирующих изображения [2]. Нейросети не справляются с определением объектов на фотографии, если внести на неё шум, а человеческий глаз не всегда в состоянии распознать, что на картинке присутствуют помехи [3].

**Актуальность работы** состоит в том, что аппаратные способы борьбы с шумами в изображениях в текущий момент не реализованы, и компании стараются бороться с дефектами на фотографиях с помощью алгоритмических методов [2]. Технические средства Kodak обрабатывают изображения с помощью таких методов, как медианный фильтр [4].

**Цель работы** — проанализировать существующие методы борьбы с шумами в изображениях.

Для достижения поставленной цели требуется решить **задачи**:

- описать термин шума, причины его появления и классифицировать его типы;
- описать существующие методы обнаружения шума в изображениях и способы его устранения;
- сформулировать критерии сравнения рассмотренных методов;
- классифицировать существующие алгоритмы борьбы с шумами.

# 1 Анализ предметной области

В данном разделе рассматриваются причины возникновения шумов в изображениях. Дается описание естественных и искусственных источников появления эффекта шума в фотографиях. Описывается состязательная атака как основной вид искусственных дефектов на изображениях. Рассматриваются различные виды естественных шумов, приводится их математическая модель. Показаны примеры шумов на изображениях в сравнении с оригиналом.

## 1.1 Причины появления шумов в изображениях

Шум – дефект изображения, в основе которого лежит эффект появления на фотографии пикселей случайного цвета на изображении [1].

Причины возникновения такого эффекта делятся на два типа: естественные и искусственные. Основным источником естественных помех на изображениях является фотосенсор [1]. Существует несколько физических объяснений появления шума на изображении [5]:

- 1) При дефектах потенциального барьера происходит утечка заряда. В этом случае шум на изображениях проявляется в виде тёмных точек на светлом фоне.
- 2) При подаче потенциала на электрод может возникнуть темновой ток, который отображается на картинке в виде светлых точек на тёмном фоне. Основная причина возникновения темнового тока — это примеси в кремниевой пластине или повреждение кристаллической решётки кремния.

Также шум на изображениях может быть вызван умышленным вмешательством человека или состязательной атакой [6]. Состязательная атака – это манипуляция обучающими данными, архитектурой модели или манипулирование тестовыми данными таким образом, что это приведёт к неправильному выходу из модели машинного обучения [6].

## 1.2 Классификация шумов

Существует несколько основных типов естественных шумов, возникающих на фотографиях [7]. От точного определения характеристики шума зависит то, какой метод требуется выбрать для автоматического определения дефектных пикселей на изображении и последующего его устранения. Для выявления причины требуется понять, на каком устройстве была сделана фотография, и через какие этапы обработки она прошла. При этом для реальных цветных фо-

тографий результирующий шум является комбинацией всех возможных типов дефектов, соответственно, нельзя точно сказать о происхождении каждого дефекта на рассматриваемой фотографии [8].

### 1.2.1 Гауссов шум

Так как квантовым процессам свойственна случайность, то такие процессы можно отнести к Гауссовым, следовательно, они обладают следующим свойством: распределение суммы независимых случайных величин сходится к нормальному, вне зависимости от характера распределения слагаемых [9].

Пусть  $I$  – интенсивность изначального пикселя, а  $\nu$  – интенсивность шума, распределённая по нормальному распределению. Тогда интенсивность загрязнённого пикселя можно представить по формуле (1.1) [7]:

$$I_f = I + \nu, \nu \sim N(0, \sigma^2). \quad (1.1)$$

Вид гауссова шума представлен на рисунке 1.1:

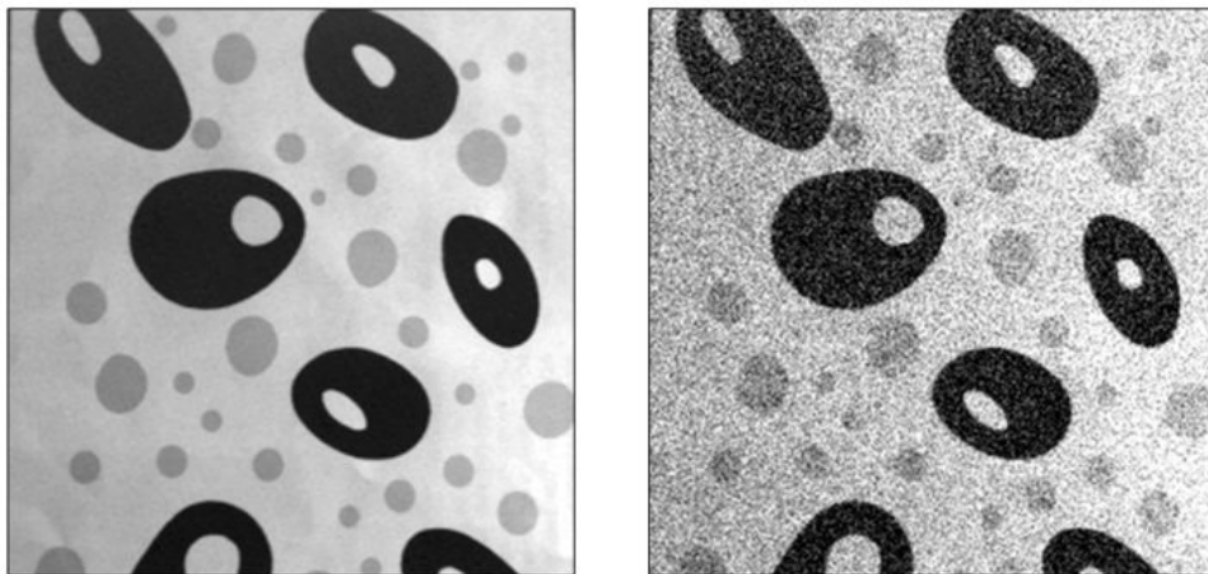


Рисунок 1.1 – Вид гауссова шума

Именно этот вид шумов на практике встречается чаще всего [7].

### 1.2.2 Шум соли и перца

Шум соли и перца проявляется в том, что на изображениях в случайных местах появляются чёрные и белые пиксели [10]. Основной причиной их возникновения является темновой ток и утечка заряда в фотосенсоре, а также на-

личие пикселей с дефектами [1].

Описать появление шума можно по формуле (1.2) [10]:

$$P(S_{i,j} = 1) = p, \quad (1.2)$$

где  $S$  – исходное изображение,  $i, j$  – координаты пикселя,  $p$  – искомая вероятность появления шума.

Вид шума соли и перца представлен на рисунке 1.2:

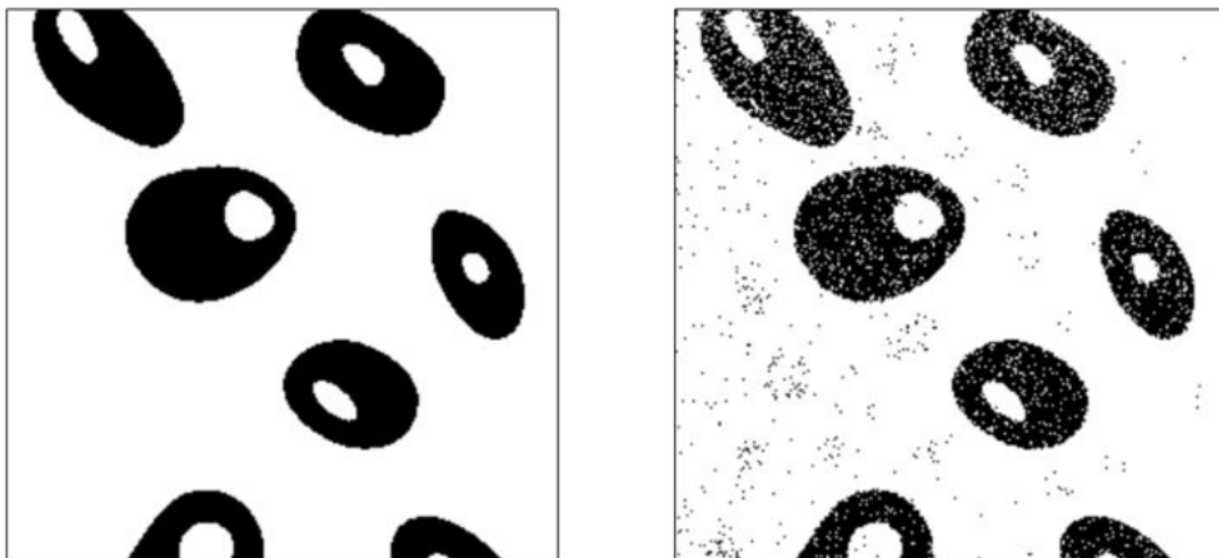


Рисунок 1.2 – Шум соли и перца

Основным методом борьбы с таким видом шумов является медианный фильтр.

### 1.2.3 Спекл-шум

Спекл-шум часто встречается в медицинских методах визуализации, которые основаны на ультразвуке и лазерных технологиях: КТ, ОКТ [11]. Сложность борьбы с такими помехами состоит в том, вероятность их возникновения описывается не нормальным распределением, а другими, например Гамма распределением или распределением Релея [12]. Влияние спекл-шума может быть значительным, а линейные методы решения таких задач не подходят для того, что исключить шумы из исходного изображения.

## 2 Обзор существующих методов

Шумы искажают исходную картинку и портят её качество так, что это способен распознать человеческий глаз. Однако могут возникнуть трудности в обнаружении помех, поскольку они трудно различимы при совпадении цвета фона и цвета пикселя, например, светлые точки будут плохо заметны на ярком фоне.

Было разработано несколько алгоритмов, которые производят бинарную классификацию пикселей и определяют, какие из них можно идентифицировать как шумы и затем их устранить.

В качестве классификации алгоритмы можно разделить на два типа:

- 1) **Изотропная фильтрация** – такие методы устраняют помехи, но не учитывают детали пикселя и увеличивают размытость.
- 2) **Анизотропная фильтрация** – алгоритмы устраняют эффекты сглаживания, уменьшают размытость и сохраняют детали пикселя, устраняя при этом непосредственно шум из изображения.

### 2.1 Общий алгоритм работы фильтров

Алгоритмы, анализирующие наличие шумов в изображениях, имеют дело с различными характеристиками одного пикселя. Например, цвет пикселя можно разбить на три составляющие – синюю, красную и зелёную [13].

В таком случае метод работает с каждой из составляющих пикселя, вычисляя новое значение для каждой характеристики. Результат работы в этом случае является объединением подсчётов по всем характеристикам.



Общая схема работы алгоритмов представлена на рисунке 2.1:

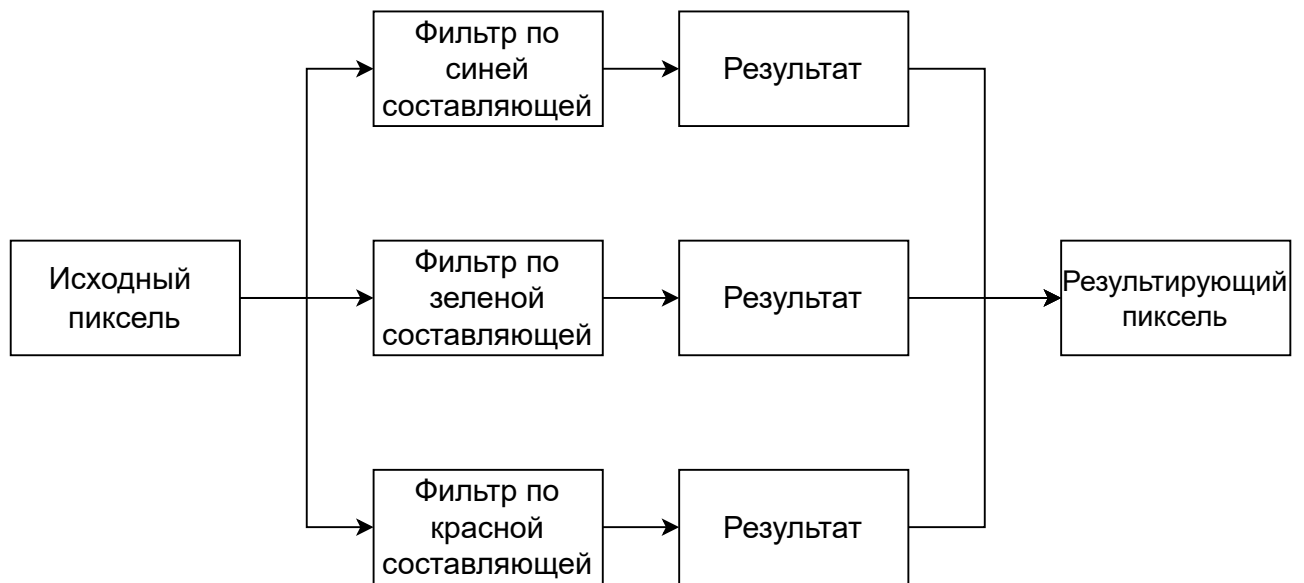


Рисунок 2.1 – Общая схема работы всех алгоритмов

Каждый из фильтров, перечисленный ниже, работает с каждым из параметров пикселя одинаково, поэтому для корректной работы алгоритмов требуется вычислить значение каждого свойства для результирующего пикселя [7].

## 2.2 Медианный фильтр

Под медианным фильтром понимается семейство однотипных алгоритмов, относящихся к классу нелинейных фильтров [14].

Метод работает в цикле с каждым пикселем изображения [7]. В окрестности каждого пикселя находится восемь соседних, каждый обладает собственными свойствами.

На рисунке 2.2 изображена сетка, с которой работает алгоритм:



Рисунок 2.2 – Рассматриваемая сетка пикселей при работе алгоритма

Пусть  $C_{i,j}$  – один из параметров рассматриваемого пикселя, а  $\Omega$  – все пиксели сетки. Алгоритм подсчитывает медиану от такого же параметра соседних клеток и заменяет параметр пикселя на значение этой медианы [14]. Итоговое значение можно посчитать по формуле (2.1):

$$C_{i,j} = \text{median}(\Omega_i). \quad (2.1)$$

Схема работы алгоритма изображена на рисунке 2.3:

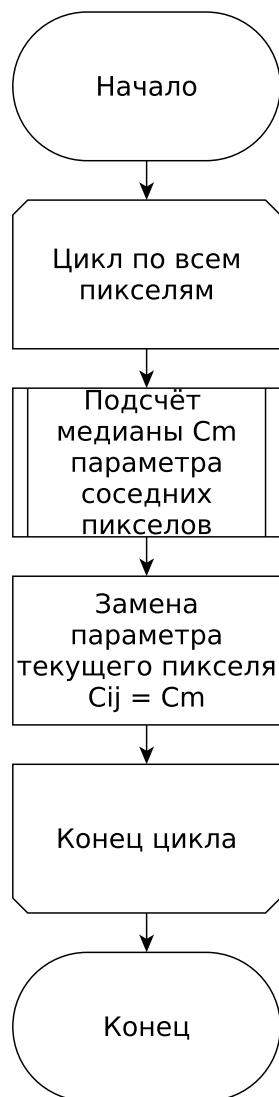


Рисунок 2.3 – Схема работы алгоритма медианного фильтра

К недостаткам данного метода можно отнести то, что алгоритм может убрать значительные детали из изображения, посчитав их за шум [15].

### 2.3 Гауссовский фильтр

Работа алгоритма гауссовского фильтра также зависит от значений цветовых свойств пикселей в сетке, рассмотренной на рисунке 2.2.

В этом случае для каждого соседнего рассчитывается вес, с которым он влияет на новое значение рассматриваемого пикселя. Пусть  $d$  – расстояние до центрального пикселя сетки,  $\sigma$  – стандартное отклонение, подсчитанное для всех значений определённого параметра текущей сетки. Тогда вес  $w$  пикселя рассчитывается по формуле (2.2) [7]:

$$w_{ij} = \exp \left( \frac{-d^2}{2\sigma^2} \right). \quad (2.2)$$

Подсчитав вес для каждого пикселя в сетки, можно рассчитать новое значение свойства рассматриваемого пикселя по формуле 2.3:

$$p_i = \frac{1}{\sum_{j \in \Omega} w_{ij}} * \sum_{j \in \Omega} w_{ij} * p_j. \quad (2.3)$$

Схема алгоритма гауссовского фильтра представлена на рисунке 2.4:



Рисунок 2.4 – Схема работы алгоритма гауссовского фильтра

## 2.4 Билатеральный фильтр

Алгоритм билатеральной фильтрации является улучшением метода Гауссовского фильтра [16].

Для каждого пикселя сетки соседних пикселей используется сразу два веса: один аналогичный параметру из исходного алгоритма, а второй отвечает за анизотропную составляющую. В методе рассчитывается разница в определённой компоненте между соседними пикселями. Если она получилась большой, то это означает, что пиксель содержит какие-то важные детали по изображению, соответственно, фильтрация приведёт к минимальным изменениям фотографии. Чем меньше разница между соседними пикселями, тем большим будет эффект от фильтрации на рассматриваемой сетке.

Расчёт веса  $w_s$ , отвечающего за изотропную составляющую, происходит по формуле (2.2). Коэффициент, регулирующий анизотропные свойства фильтрации, рассчитывается по формуле (2.4) [16]:

$$w_r = \exp \left( \frac{-|p_i - p_j|}{2\sigma^2} \right). \quad (2.4)$$

В таком случае результат работы некоторого пикселя можно посчитать по формуле (2.5):

$$p_i = \frac{1}{\sum_{j \in \Omega} w_s w_r} * \sum_{j \in \Omega} w_s w_r p_j. \quad (2.5)$$

Схема работы алгоритма представлена на рисунке 2.5:



Рисунок 2.5 – Схема работы алгоритма билатеральной фильтрации

## 2.5 Алгоритм Цзяньвэй

Этот алгоритм был описан в 2014 году индонезийским учёным Ван Цзяньвэй [17]. Алгоритм позволяет эффективно убирать шумы соли и перца даже в случае сильной загрязнённости изображения. Метод предполагает, что для

каждого пикселя помехи будут удалены по всем цветовым составляющим.

Процедура заключается в обходе всех пикселей фотографии в заданном порядке и определении того, соответствуют ли значения пикселей функции плотности вероятности импульсного шума или нет. Если пиксель на первом классифицируется как шум, то подсчитывается количество импульсного шума в маске определенной формы. Если это число меньше чем заданный порог, то пиксель рассматривается как возможный шум. Результатом операции маски является замена значения пикселя. В противном случае это не рассматривается как шум, значение пикселя остается неизменным.

Схема алгоритма Цзяньвэй представлена на рисунке 2.6:

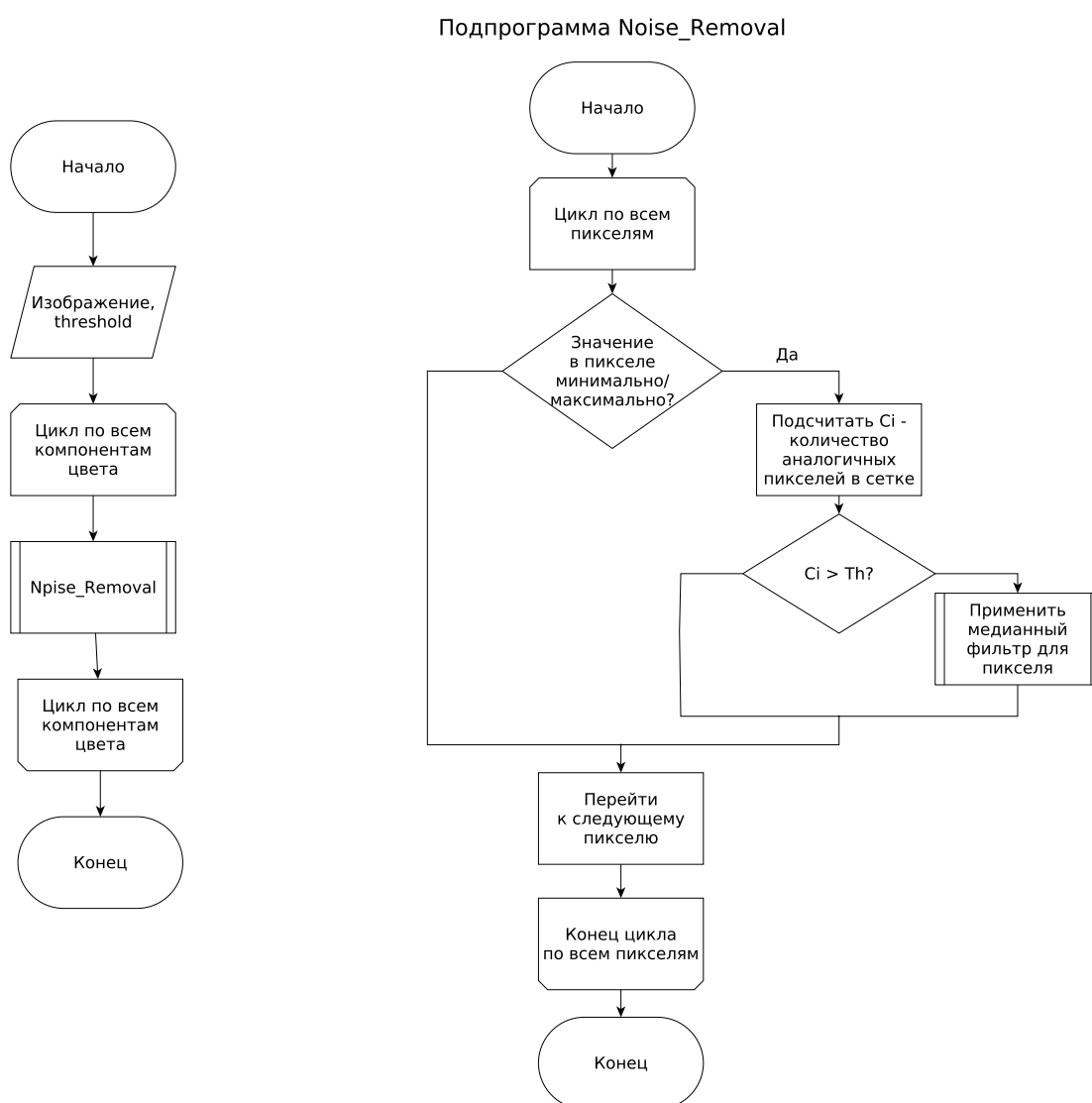


Рисунок 2.6 – Схема работы алгоритма Цзяньвэй

## 2.6 DnCNN

Алгоритмы, основанные на сверточных нейронных сетях, получили широкое распространение в области удаления шумов из изображений. Они применяются в ситуациях, требующих восстановления исходного изображения по имеющемуся загрязненному фото, например, при анализе снимков из телескопов [18]. DnCNN относится к этому классу методов.

Перед запуском алгоритма требуется задать суммарную глубину слоев  $D$ , а также количество цветовых составляющих  $c$  – для цветных изображений  $c = 3$ , для серых –  $c = 1$  [19]. Всего в конфигурации нейронной сети используется три типа слоев [20]:

- 1) Первый слой состоит из 64 фильтров размером  $3 \times 3 \times c$ , в качестве функции активации используется ReLU.
- 2) Следующие  $2 * (D - 1)$  слоев состоят из фильтров размером  $3 \times 3 \times 64$ , между каждым слоем применяется пакетная нормализация.
- 3) Последний слой состоит из  $c$  фильтров размером  $3 \times 3 \times 64$  каждый.

Во всех слоях, кроме последнего, используется ReLU в качестве функции активации нейрона [19].

Пусть  $x$  – входное значение нейрона.

Тогда выход считается по формуле (2.6):

$$f(x) = \max(0, x). \quad (2.6)$$

Конфигурация нейронной сети представлена на рисунке 2.7:

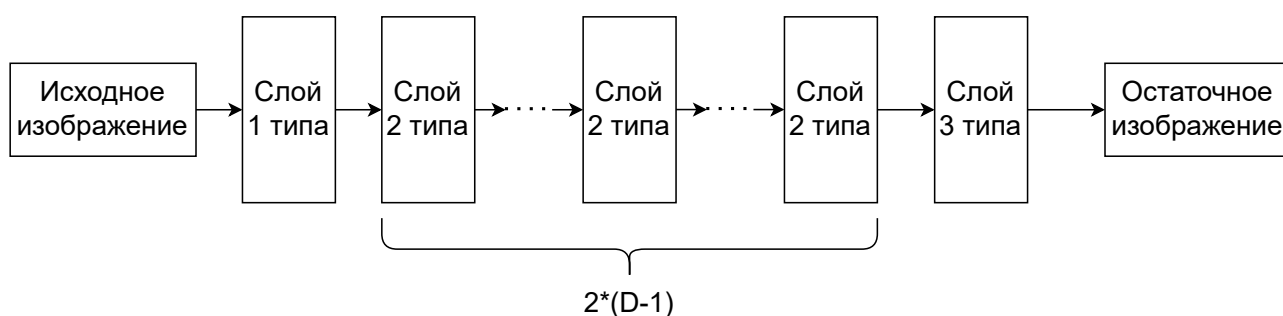


Рисунок 2.7 – Конфигурация нейронной сети в алгоритме DNCNN

На выходе алгоритма получается остаточное изображение  $R$ . Пусть  $Y$  – исходное изображение. Тогда очищенный снимок  $X$  можно посчитать по формуле (2.7):

$$X = Y - R. \quad (2.7)$$

## 2.7 RIDNet

Алгоритм RIDNet также использует сверточные нейронные сети, однако в его основе лежит другой подход [21]. Идея состоит в том, что не во всех ситуациях требуется, чтобы модель одинаково оценивала каждую цветовую составляющую пиксела, поэтому у каждой из составляющих будет свой вес [22].

Конфигурация сети состоит из трех основных модулей [22]:

- 1) **Модуль извлечения признаков** – он состоит из одного слоя, который производит свертку изображения и инициализирует начальные признаки  $f_0$ .
- 2) **Модуль извлечения остаточных признаков из остаточного изображения** – выполняет основную работу в сети, состоит из четырех дополнительных модулей – EAM (Enhancing attention module).
- 3) **Модуль реконструкции исходного изображения** – использует один слой, восстанавливает исходное изображение. ReLU используется в качестве функции активации.

При этом модули в алгоритме связаны между собой, например, после прохождения первого модуля полученный результат попадает на вход как второго модуля, так и третьего.

Общая конфигурация нейронной сети представлена на рисунке 2.8:

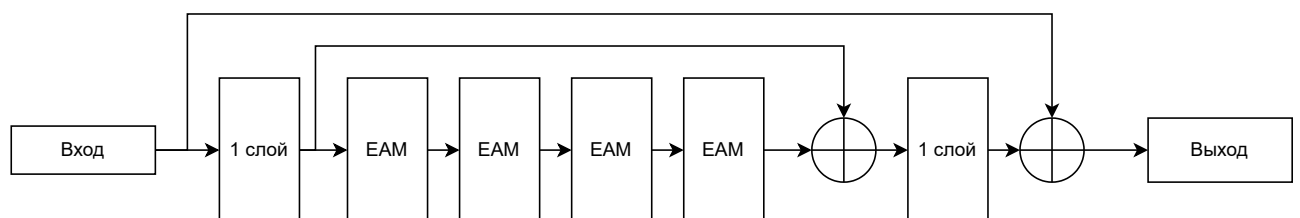


Рисунок 2.8 – Общая конфигурация нейронной сети в алгоритме RIDNet

EAM внутри себя состоит из множества различных слоев, которые комбинируются между собой [22]. Каждый модуль внутри себя выполняет три действия: выделяет основные признаки из изображения, сжимает изображение для



большей производительности и увеличивает веса по наиболее значимым признакам.

Внутри модуля используются сетки размером  $3 \times 3$ . В качестве функции активации на выходе последнего слоя ЕАМ используется сигмоида. Пусть  $x$  – вход нейрона, тогда выход можно посчитать по формуле (2.8):

$$f(x) = \frac{1}{1 + e^x}. \quad (2.8)$$

При этом авторы алгоритма предложили использовать больше, чем четыре ЕАМ-модуля, однако это не увеличило результаты работы алгоритма [21]. Также в алгоритме предусмотрено использование регуляризации для борьбы с переобучением системы.

### 3 Критерии сравнения алгоритмов

Для верификации работы алгоритмов используются открытые наборы реальных изображений. Алгоритмам предоставляют на вход загрязненные изображения, в результате получается очищенное изображение. У начальных изображений зафиксирован одинаковый размер, так как это позволяет зафиксировать размер входа для нейронной сети. Для полученного результата подсчитываются метрики, и на их основе устанавливается эффективность для метода.

Для сравнения результатов используется две основные метрики: PSNR и SSIM [23].

#### 3.1 PSNR

PSNR – метрика, обозначающее пиковое отношение сигнала к шуму [24]. Она универсальна и используется не только для удаления шумов, а также для измерения уровня искажения при сжатии изображений. PSNR рассчитывается по логарифмической шкале.

Пусть есть два изображения  $I$  – исходное, и  $K$  – полученное в результате обработки. Размер каждого составляет  $M \times N$  пикселей.

Разница между ними рассчитывается по формуле (3.1) [24]:

$$\text{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I(i, j) - K(i, j)|. \quad (3.1)$$

Максимальное значение любого пикселя обозначим за  $MX$ . Чаще всего оно равно 255 [24]. Теперь искомую метрику PSNR можно рассчитать по формуле (3.2):

$$\text{PSNR} = 20 \log_{10} \left( \frac{MX}{\sqrt{\text{MSE}}} \right). \quad (3.2)$$

Метрика изначально была создана для монохромных изображений, поэтому для цветных изображений результат усредняется по каждой из цветовых составляющих пикселя.

#### 3.2 SSIM

SSIM – метрика, обозначающая индекс структурного сходства изображений [25]. В отличие от PSNR, эта характеристика позволяет учесть не только разницу в фактических величинах между пикселями, но и взаимосвязь между ними, то есть структурное сходство.

Пусть есть два изображения  $I$  – исходное, и  $K$  – полученное в результате обработки. Размер каждого составляет  $M \times N$  пикселей. Для каждого изображения были рассчитаны характеристики:  $\mu_i$  – среднее по цветовой составляющей для изображения  $i$ ,  $\sigma_i^2$  – дисперсия цветовой составляющей для фотографии  $i$ ,  $\sigma_{ij}$  – ковариация цветовой составляющей для изображений  $i$  и  $j$ .

Также вводится константа  $L$ , которая обозначает динамический диапазон. Как правило, она равняется 255 [25].

Исходя из этого, метрику SSIM можно рассчитать по формуле (3.3) [25]:

$$\text{SSIM} = \frac{(2\sigma_I^2\sigma_K^2 + 0.01L)(2\sigma_{IK}^2 + 0.03L)}{(\mu_I^2 + \mu_K^2 + 0.01L)(\sigma_I^2 + \sigma_K^2 + 0.03L)}. \quad (3.3)$$

## 4 Классификация существующих решений

Классификацию существующих алгоритмов можно привести к таблице. В качестве основных параметров классификации были использованы следующие признаки:

- учитывается ли анизотропная составляющая при очистке изображения от шумов;
- учитывают ли алгоритмы взаимосвязь пикселей между собой;
- используются ли алгоритмом нейронные сети;
- происходит ли сглаживание в ходе работы алгоритма;
- какой размер изображения требуется для его работы;
- для какого вида шумов используется алгоритм.

Классификация существующих алгоритмов представлена на таблице 1. Под заголовком Gauss приведены критерии для алгоритма гауссовского фильтра, под заголовком Median – для алгоритма медианного фильтра, Bilat – для билатерального фильтра, Jianwei – для алгоритма Цзяньвэй.

Таблица 1 – Таблица сравнения существующих алгоритмов

	Median	Gauss	Bilat	Jianwei	DnCNN	RIDNet
Учитывается ли анизотропия?	нет	нет	да	да	да	да
Учитывается ли взаимосвязь пикселей?	нет	да	да	да	нет	да
Происходит ли сглаживание?	нет	да	да	да	нет	нет
С каким размером фото работает метод?	любой	любой	любой	любой	512x512	512x512
С каким шумом работает метод?	соли и перца	гауссов	гауссов	соли и перца	любой	любой
Используются нейронные сети?	нет	нет	нет	нет	да	да

## ЗАКЛЮЧЕНИЕ

Была проанализирована предметная область, а также рассмотрены основные алгоритмы борьбы с шумами в изображениях. Приведены критерии сравнения алгоритмов и осуществлена их классификация.

Полученные результаты можно свести к следующим тезисам:

- шум на изображениях возникает вследствие несовершенства технических средств. На практике применяются алгоритмические средства борьбы;
- существующие алгоритмы, не использующие нейронные сети, справляются лишь с определенным видом шумов, имея проблемы с устранением остальных типов;
- методы, использующие сверточные нейронные сети, работают лишь с определенным размером входного изображения, при этом не производя сглаживание;
- в алгоритмах DnCNN и RIDNet детали реализации сети могут быть изменены: функция активации, количество слоев;
- новые решения в области удаления шумов являются лишь улучшением или комбинированием уже существующих алгоритмов.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Thanh Hien Nguyen. Image Noise Removal Method Based on Thresholding and Regularization Techniques // IEEE Access. 2022. С. 71584–71597.
2. H.W. Khushi. Impulse Noise Removal Using Soft-computing // Lahore Garrison University Research Journal of Computer Science and Information Technology. 2022. С. 32–48.
3. M.L. Nguyen. BoostNet: A Boosted Convolutional Neural Network for Image Blind Denoising // IEEE Access. 2021.
4. Rabie T. Adaptive Median Filtering of CCD Sensor Noise. 2005.
5. Зотов П.В. Цифровой шум изображения и его прикладное значение в криминалистике // Вестник СГЮА. 2015.
6. Голдобин И.А. Климова Е.И. Влияние шумов на алгоритмы цифровой обработки изображений // Актуальные вопросы развития современной цифровой среды. 2021. С. 396–402.
7. Hambal Abdalla Mohamed. Image Noise Reduction and Filtering Techniques // International Journal of Science and Research (IJSR). 2017. С. 2033–2037.
8. H.W. Xiangming. An improved nonlocal means-based correction strategy for mixed noise removal // IET Image Processing. 2022.
9. Ćeranić Š.T. Noise reduction quality test for two-photon laser scanning microscopic images // IEEE Access. 2022.
10. G.M. Ganesan. Noise Detection in Images using Moments // Research Journal of Applied Sciences, Engineering and Technology. 2015. С. 307–314.
11. W.K Wenjie. Speckle noise removal based on structural convolutional neural networks with feature fusion for medical image // Signal Processing: Image Communication. 2021.
12. Yan Nao Siwei. Speckle Noise Removal Model Based on Diffusion Equation and Convolutional Neural Network // Computational Intelligence and Neuroscience. 2022. С. 1–11.

13. A. Bandyopadhyay. High Density Impulse Noise Removal from Color Images by K-means Clustering based Detection and Least Manhattan Distance-oriented Removal Approach // International Journal of Advanced Computer Science and Applications. 2021. C. 608–614.
14. S. Khan. An adaptive dynamically weighted median filter for impulse noise removal // EURASIP Journal on Advances in Signal Processing. 2017.
15. Kunsoth R. Biswas M. Modified decision based median filter for impulse noise removal. 2016. C. 1316–1319.
16. M. Nabahat. Optimization of bilateral filter parameters using a whale optimization algorithm // Research in Mathematics. 2022.
17. Wang Jianwei. A Noise Removal Algorithm of Color Image // TELKOMNIKA Indonesian Journal of Electrical Engineering. 2014.
18. M. Fujimoto. Noise Removal for Degraded Images Using Neural Networks // IEEE Transactions on Electronics, Information and Systems. 2002. C. 1301–1308.
19. V. Murali. Image Denoising Using DnCNN: An Exploration Study // IEEE Access. 2020. C. 847–859.
20. K. Zhang. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising // IEEE Transactions on Image Processing.
21. N. Loubere. RiDNet Practical Fieldwork Notes. 2012.
22. X. Li. RIDNet: Recursive Information Distillation Network for Color Image Denoising. 2019. C. 3896–3903.
23. D.M. Setiadi. PSNR vs SSIM: imperceptibility quality assessment for image steganography // Multimedia Tools and Applications. 2021.
24. M. Nouman. Evaluation of Image Super Resolution Deep Learning Technique based on PSNR Value. 2022. C. 93–122.
25. V. Subramanian. SSIM Compliant Modeling Framework With Denoising and Deblurring Applications // IEEE Transactions on Image Processing. 2021.

## **ПРИЛОЖЕНИЕ А**