

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Факультет математики, механики и компьютерных технологий
Кафедра прикладной математики и программирования
Направление подготовки: 01.03.02 Прикладная математика и информатика

РАБОТА ПРОВЕРЕНА

Рецензент,

_____/_____
«__»_____ 2019г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
доцент

_____/А.А. Замышляева
«__»_____ 2019 г.

Разработка и исследование алгоритмов восстановления поврежденных
и изображений с низким разрешением

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ–01.03.02.2019.57. ПЗ ВКР

Руководитель работы, доцент
_____/Т.Ю. Оленчикова

«__»_____ 2019 г.

Автор работы

Студент группы ЕТ-412

_____/ А.В. Махмутов

«__»_____ 2019 г.

Нормоконтролер, ассистент

_____/ Н.С.

Мидоночева

«__»_____ 2019 г.

Челябинск
2019

АННОТАЦИЯ

Махмутов А.В. Разработка и исследование алгоритмов для восстановления поврежденных и изображений с низким разрешением. – Челябинск: ЮУрГУ, ЕТ-412, 51 с., 26 ил., 7 табл., библиогр. список – 18 наим., 1 прил.

Цель данной работы – разработка и исследование алгоритмов восстановления поврежденных и изображений с низким разрешением.

В первом разделе были рассмотрены методы для детектирования повреждений, устранения дефектов и сверточные нейронные сети для восстановления качества сжатых изображений.

Во втором разделе была описана математическая модель устранения повреждений и модель сверточной нейронной сети.

Третий раздел посвящен разработке алгоритмов восстановления поврежденных изображений, архитектуры сверточной нейронной сети.

Четвертый раздел содержит экспериментальные данные. В данном разделе были выбраны метрики качества и дана оценка результатов работы и модели сверточной нейронной сети.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	7
1 МЕТОДЫ ВОССТАНОВЛЕНИЯ ПОВРЕЖДЕННЫХ ИЗОБРАЖЕНИЙ И ИЗОБРАЖЕНИЙ С НИЗКИМ РАЗРЕШЕНИЕМ	10
1.1 Требования к программе для восстановления изображений.....	10
1.2 Детектирование	10
1.2.1 Гистограмма яркости изображения	10
1.2.2 Оператор локальных бинарных окрестностей.....	11
1.3 Методы для восстановления изображений	12
1.3.1 Метод Шепарда	12
1.3.2 Метод Криминиси.....	13
1.3.3 Сверточные нейронные сети	15
1.3.4 Модель супер-разрешения	16
1.3.5 Модель устранения артефактов.....	17
1.4 ПО для восстановления изображений	18
1.4.1 AKVIST Retoucher	18
1.4.2 JPEG Recovery Pro.....	19
1.5 Постановка задачи	20
1.6 Выводы по первому разделу	21
2 МОДЕЛИ ВОССТАНОВЛЕНИЯ ИЗОБРАЖЕНИЙ	22
2.1 Детектирование поврежденных участков изображения	22
2.1.1 Метод локальных бинарных окрестностей	22
2.1.2 Метод опорных векторов	24
2.2 Метод интерполяции для восстановления изображения	26
2.3 Модели сверточной нейронной сети для восстановления сжатых изображений	27
2.4 Выводы по второму разделу	31
3 РАЗРАБОТКА АЛГОРИТМОВ ВОССТАНОВЛЕНИЯ ИЗОБРАЖЕНИЙ .	32
3.1 Алгоритм предварительной обработки изображения	32

3.2 Алгоритм LBP	32
3.3 Алгоритм создания бинарной маски структуры изображения	34
3.4 Алгоритм классификации	35
3.5 Алгоритм восстановления изображения	36
3.6 Структура сверточной нейронной сети	38
3.7 Разработка модуля восстановления изображений.....	39
3.7.1 Диаграмма компонентов	39
3.7.2 Описание библиотек модуля	39
3.8 Выводы по разделу	40
4 ИССЛЕДОВАНИЕ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ	41
4.1 Метрики оценки эффективности алгоритмов	41
4.2 Оценка результатов.....	42
4.2.1 Результат работы алгоритма восстановления изображения.....	42
4.2.2 Результат работы сверточной нейронной сети	44
4.2.3 Результаты обучения	45
4.3 Выводы по разделу	46
ЗАКЛЮЧЕНИЕ	48
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	50
ПРИЛОЖЕНИЕ 1. Исходный код программы.....	52

ВВЕДЕНИЕ

В современном мире существует огромное количество компьютерных технологий, без которых невозможно представить нашу повседневную жизнь. Создается все больше различных программ, каждая из которых выполняет свою определенную задачу.

Задача компьютерной графики заключается в обработке информации, связанной с изображениями. Существует несколько направлений графики: обработка изображений, распознавание изображений, визуализация. Обработка изображений – является актуальной проблемой в наше время. Задачей обработки может быть, как улучшение изображения по какому-либо определенному критерию, например, устранение различных шумов на старых фотографиях, так и преобразования, изменяющие изображение полностью.

В данной работе исследованы методы, различные подходы к обработке и восстановлению изображений, устранению дефектов на изображениях, облегчающие труд человека, а также, способные выполнять такие интеллектуальные задачи, которые не способен выполнить обычный человек. В связи с этим задача обработки и восстановления изображений остается актуальной.

Цель данной работы – реализовать методы восстановления поврежденных изображений с использованием математических алгоритмов и сверточных нейронных сетей.

Для достижения поставленной цели необходимо решить следующие задачи:

- разработать математические модели устранения дефектов;
- разработать и реализовать алгоритмы устранения дефектов типа царапин и импульсных шумов;
- разработать сверточную нейронную сеть для восстановления изображений с низким разрешением;

– разработать методику экспериментальной проверки качества предложенных алгоритмов;

– разработать программу и выполнить обработку результатов эксперимента.

Данная работа направлена на устранение царапин на изображениях, улучшение качества сжатых изображений и восстановление их до состояния, максимально приближенного к оригинальному изображению. В конечном итоге, результат программы – это изображение с устраненными повреждениями и восстановленным качеством.

В работе рассматриваются различные способы детектирования повреждений в виде царапин и импульсных шумов, а также основные методы интерполяции для восстановления детектированных поврежденных участков. Каждый метод имеет свои достоинства и недостатки, алгоритм работы и область применения у каждого метода абсолютно разная. Для исследования были выбраны лучшие методы детектирования: оператор локальных бинарных окрестностей [7] и метод опорных векторов [8], которые в дальнейшем были доработаны.

Для восстановления поврежденных участков использовался метод интерполяции (метод Телеа [17]), который, согласно литературным источникам, показывал результаты намного лучше, чем другие методы. Для восстановления качества сжатых изображений рассматривались сверточные нейронные сети. Были представлены модели рассматриваемых сверточных нейронных сетей. Как оказалось, качество восстановления зависит не от количества и типа изображений в обучающей выборке, а от структуры самой сверточной нейронной сети. Одна модель решает проблему сильно сжатых изображений, другая модель восстанавливает слабо сжатые изображения, но качество выходного изображения настолько приближено к оригинальному, что разница не заметна человеку. На основе данных моделей в работе разработана и реализована сверточная нейронная сеть, которая является более стабильной и способной восстанавливать как сильно сжатые изобра-

жения, так и слабо сжатые изображения. Человеку порой бывает трудно визуально определить разницу между восстановленным и оригинальным изображением, если уровень повреждений небольшой. Для этого в работе были использованы различные метрики качества оценки восстановления изображений. Так же данные метрики использовались для сравнения моделей сверточных нейронных сетей, так как оценить результат визуально практически невозможно.

1 МЕТОДЫ ВОССТАНОВЛЕНИЯ ПОВРЕЖДЕННЫХ ИЗОБРАЖЕНИЙ И ИЗОБРАЖЕНИЙ С НИЗКИМ РАЗРЕШЕНИЕМ

1.1 Требования к программе для восстановления изображений

Сформулируем несколько требований для того, чтобы программа для восстановления изображений применялась в реальном мире и могла справляться с различными дефектами изображений, а также улучшать качество изображений с низким разрешением.

Данная разработка должна:

- устранять различного рода царапины, например, на старых фотографиях;
- улучшать качество изображений с низким разрешением;
- обрабатывать и восстанавливать изображения с другими различными импульсными шумами;
- устранять артефакты сжатия.

1.2 Детектирование

1.2.1 Гистограмма яркости изображения

Существуют ручные, полуавтоматические и автоматические способы детектирования поврежденных участков [1, 2]. В данной работе нас интересуют автоматические методы детектирования [3]. Существует много алгоритмов по нахождению царапин и импульсных шумов [4, 5]. Одним из методов автоматического детектирования повреждений является способ, использующий гистограмму яркости изображения [6]. На гистограмме дефект или другой шум – как правило, сильно выделяющаяся линия, которая нарушает однородность фона изображения. Большое или маленькое значение пикселя по отношению к соседним пикселям в большинстве случаев является дефектом. Данный метод не учитывает другие структурные особенности, плохо находит импульсные шумы. Так же данный способ непригоден для

нахождения повреждений разных размеров, так как необходимо задавать пороговые значения. Наиболее существенным недостатком метода гистограмм яркости является то, что метод за дефект может принять и часть изображения.

1.2.2 Оператор локальных бинарных окрестностей

Для обнаружения шумов и повреждений на изображении используется оператор локальных бинарных окрестностей (LBP) [7]. Хотя оператор и применяется в большинстве случаев для распознавания и полученные LBP оператором данные обычно подают на вход нейронной сети, он так же используется в задачах по работе с текстурами. С помощью дескрипторов локальных бинарных окрестностей получается гистограмма изображения, которая описывает структуру изображения. LBP быстро вычисляется в программах, так как работает только с целыми числами.



Рисунок 1.1 – Пример работы LBP

1.3 Методы для восстановления изображений

1.3.1 Метод Шепарда

В качестве более общего решения, подходящего для подавления, как импульсных шумов, так и обычных царапин, используются метод Шепарда, использующий некоторую окрестность пикселя для определения вида и корректировки шума, присущего центру этой окрестности [8].

Так выглядит функция Шепарда для интерполяции:

$$\frac{F(x)}{\sum_{x \in N} 1} \quad (1.1)$$

где $F(x)$ – сглаживаемая часть изображения,

x – поврежденные пиксели,

N – окрестность поврежденного пикселя.

С помощью данного метода можно так же убирать царапины на старых архивных фотографиях (рисунок 1.2).

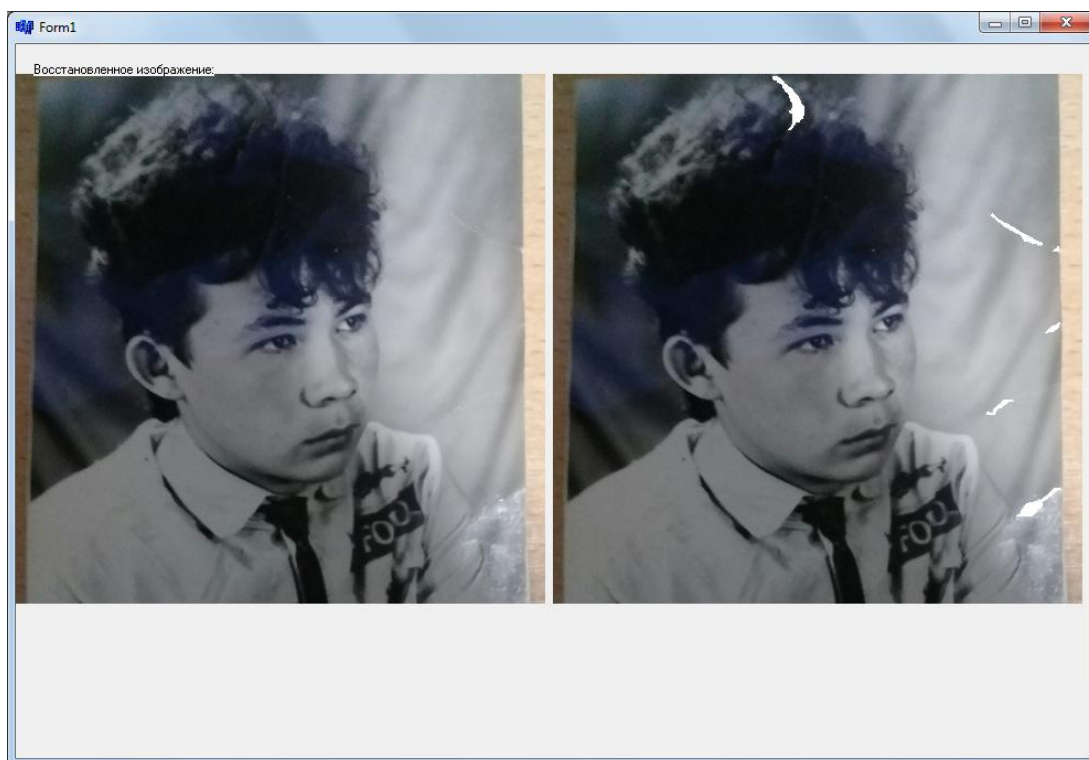


Рисунок 1.2 – Устранение царапин методом Шепарда

Главным недостатком метода является то, что он недостаточно хорошо закрашивает поврежденные участки. Цвет поврежденного участка получается однотонным, что делает восстановленный участок заметным человеческому зрению.

1.3.2 Метод Криминиси

Алгоритм, разработанный Criminisi [9, 10], довольно сильно отличается от предыдущего метода. Данный метод не использует диффузию, а блоками восстанавливает неизвестную область изображения. Экземпляры для восстановления находятся из известных и схожих областей изображения. Алгоритм способен восстанавливать как небольшие шумы, так и значительные повреждения, сохраняя при этом цветовую гамму изображения и сводя потерю качества изображения к минимуму, заполняя поврежденную площадь схожими, известными областями исходного изображения. Но для этого необходимо использовать начальную окрестность большого размера для больших повреждений и соответственно уменьшать ее для небольших повреждений, что требует ручных корректировок. Так же данный алгоритм плохо сохраняет границы и может ухудшить не поврежденные части изображения.

Алгоритм работы. Размер Ψp (блока) вводится вручную. Приоритет каждого блока вычисляется по формуле:

$$(1.2)$$

где $C(p)$ – множитель уверенности,

$D(p)$ – множитель данных.

Они вычисляются следующим образом:

$$(1.3)$$

где I – изображение,

Ω – область сегмента ,

n_p – нормаль, ортогональная границе Ω ,

– представляет собой вектор изофот,

α – параметр нормализации,

$D(p)$ – отвечает за геометрическую составляющую.

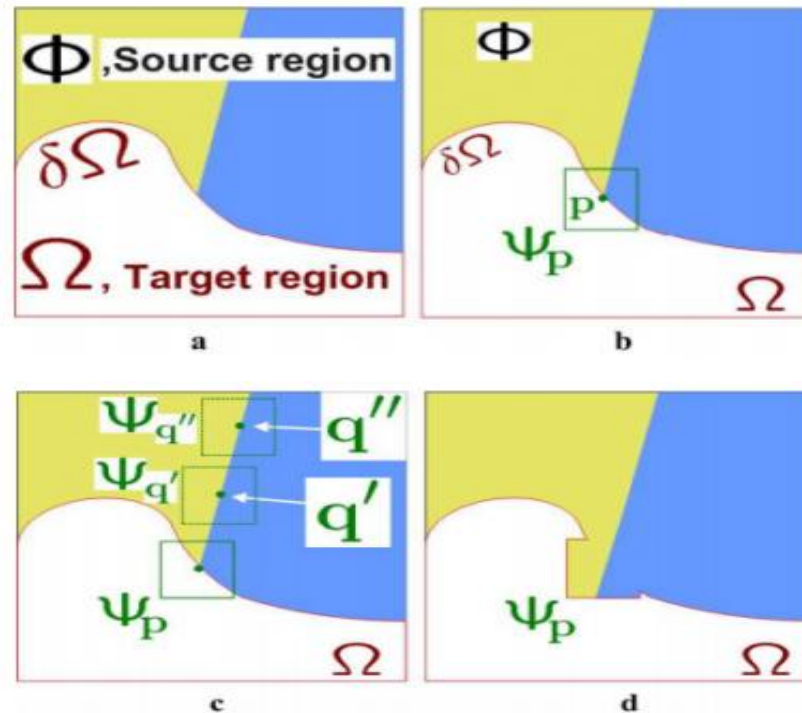


Рисунок 1.3 – Алгоритм Криминиси

На рисунке 1.3 часть (а): – область, которую необходимо восстановить, – известная область изображения, а – граница области . Часть (b): задача состоит в соединении области, разделенной некоторым сегментом , в точке \in . Часть (c): подходящие кандидаты лежат на границе между двух текстур в . В качестве примера указаны области и . Часть (d): выбирается наиболее подходящий блок и копируется в блок . Таким образом, достигается заполнение части области , а описывает зависимость от окружающих пикселей. Если пикселей из области будет больше в области точки , то будет принимать наивысшее значение. В частности, начальные условия следующие:

Далее выбирается сегмент, наиболее подходящий по приоритету.

1.3.3 Сверточные нейронные сети

Наилучшие результаты в области обработки изображений показывают сверточные нейронные сети. Данные нейронные сети обладают устойчивостью к масштабированию, смене ракурса и прочим искажениям [11].

Сверточные нейронные сети состоят из нескольких видов слоев:

- сверточный слой;
- субдискретизирующий слой;
- полносвязный слой.

Сверточный слой может быть описан следующим выражением:

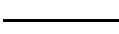
(1.4)

где S – выход свертки,

I – входной двумерный сигнал,

K – ядро свертки.

Формула (1.4) описывает движение ядра по входному изображению (рисунок 1.4). После использования операции свертки к выходу нейронной сети применяется нелинейная функция активации. Основное назначение функции активации – сделать нейронную сеть нелинейной. Часто используемыми функциями активации являются:

- сигмоидная: 
- гиперболический тангенс:
- линейная
- положительно-линейная (ReLU):

Слой субдискретизации (Pooling layer) представляет собой нелинейное уплотнение группы входных значений до одного значения. Например, слой выборки максимального значения (max pooling) вычисляет максимальное значение входов в прямоугольной области (рисунок 1.5).

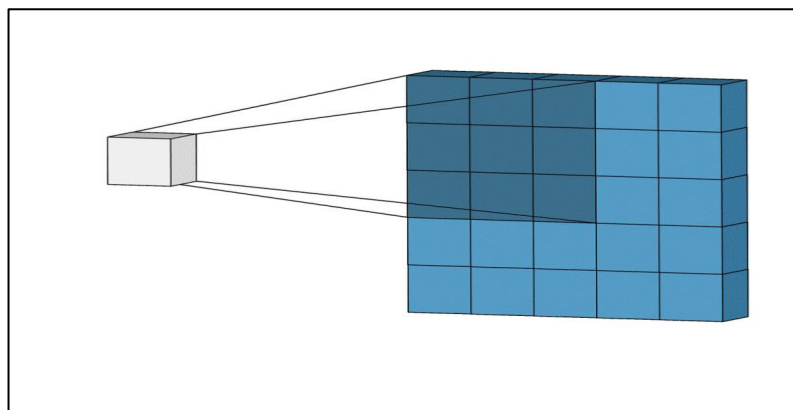


Рисунок 1.4 – Ядро свертки

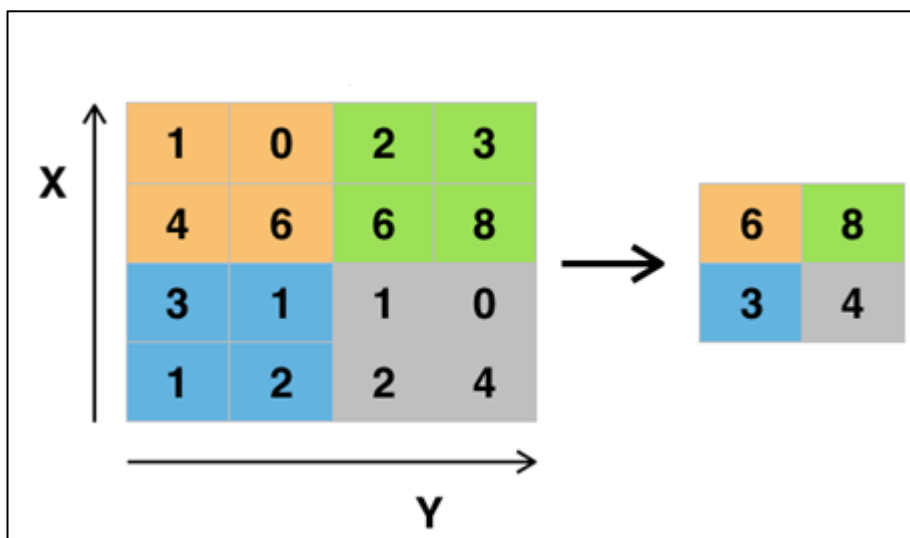


Рисунок 1.5 – Слой макспулинга

Другие популярные слои субдискретизации либо вычисляют среднее значение в области, либо взвешенное среднее.

После свёрточных и субдискретизирующих слоев мы получаем множество карт признаков. Их соединим в один вектор, и этот вектор подаем обычной нейронной сети – полносвязный слой.

К сожалению, сверточные нейронные сети являются универсальным методом в устранении повреждений и не могут давать результаты лучше частных методов.

1.3.4 Модель супер-разрешения

Сверточная нейронная сеть супер-разрешения (Super Resolution Convolution Neural Network, SR CNN) реализует сквозное отображение, которое принимает изображение с низким разрешением в качестве входа,

затем выдаёт на выход изображение с высоким разрешением [12]. Сеть содержит три сверточных слоя, каждый из которых отвечает за конкретную задачу. Первый слой извлекает признаки из входного изображения и представляет их в качестве многомерной матрицы. Затем слой нелинейного отображения отображает каждую многомерную матрицу первого слоя на другую многомерную матрицу. Слой восстановления объединяет матрицы для создания конечного результата. Архитектура сети довольно простая: первый свёрточный слой размером ядра 9×9 и с количеством каналов 64, второй свёрточный слой размером фильтра 1×1 и числом каналов 16, и последний слой размером фильтра 5×5 и с количеством каналов 1 (рисунок 1.6). Данная модель сверточной нейронной сети отлично справляется с не сильно сжатыми изображениями. При сильно сжатом изображении на изображении появляются другие различные шумы в виде колец. Поэтому данная модель не совсем подходит для данной работы.

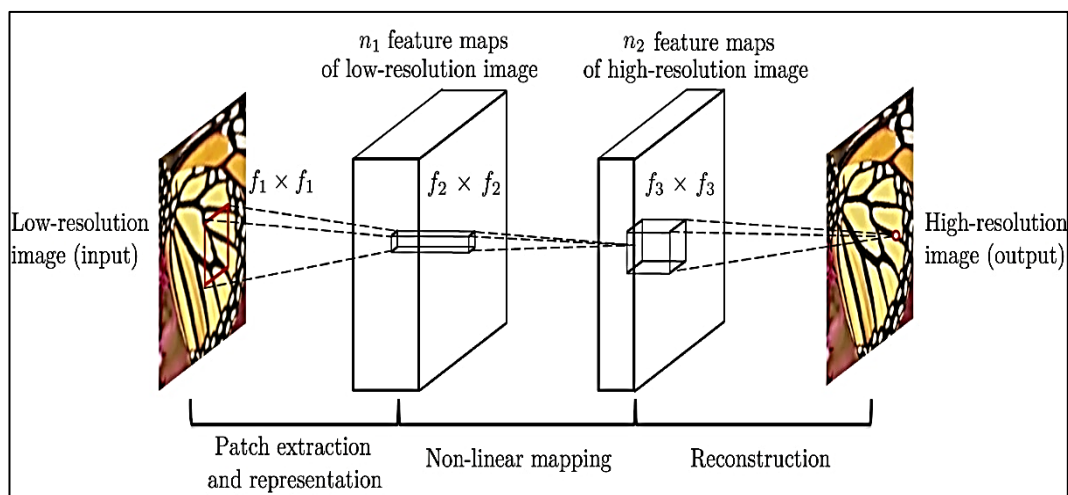


Рисунок 1.6 – Модель SR CNN

1.3.5 Модель устранения артефактов

Сверточная нейронная сеть для уменьшения артефактов (Artefact Reduction Convolution Neural Network, AR CNN) создана в 2015 году на основе модели SR CNN и в большинстве случаев данная модель показывает результаты лучше, чем предыдущая модель [13, 14]. В модели Artifact Reduction Convolution Neural Network первый этап – извлечение признаков –

определяет, что следует восстанавливать на следующих этапах. Размер ядра на данном слое 9 9. В отличие от прошлой модели добавлен дополнительный слой, который позволяет улучшить признаки, извлекаемые из изображения. Размер ядра данного свёрточного слоя 7 7, количество фильтров 32. В модели ARCNN не используется слой субдискретизации и полносвязный слой, поэтому конечный выход изображения имеет тот же размер, что и входное изображение [18]. В качестве функции активации используется положительно-линейная функция.

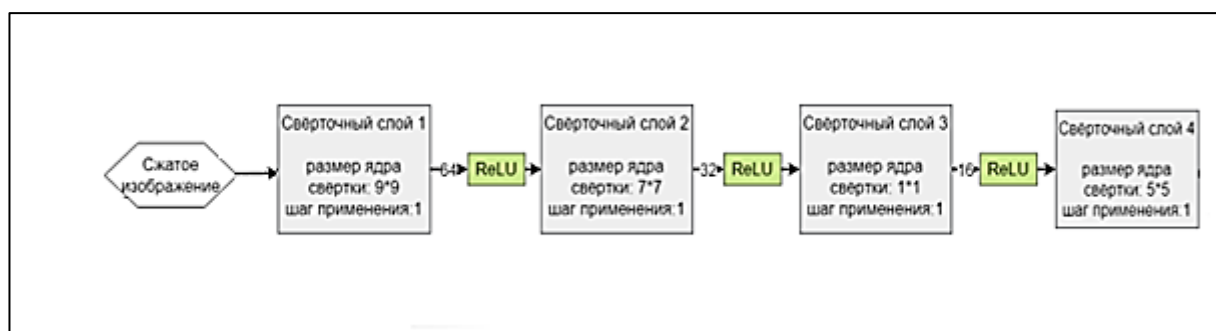


Рисунок 1.7 – Архитектура сети ARCNN

Модель справляется со всеми видами сжатия, в том числе и с изображениями, сжатыми алгоритмом JPEG, однако выходное качество изображения при небольших сжатиях показывает результаты хуже, чем предыдущая модель супер-разрешения, но работает лучше при сильном сжатии.

1.4 ПО для восстановления изображений

1.4.1 AKVIST Retoucher

AKVIST Retoucher устраняет дефекты с поверхности фото, царапины, пыль. В данной разработке применяется алгоритм диффузии, он заполняет недостающие фрагменты, путем сглаживания, используя соседние участки некоторой известной окрестности изображения. Программа эффективно дорисовывает поврежденные участки фотографий. Так же данная разработка может удалять какие-либо объекты (машин, людей, случайно попавших в кадр) на изображениях, текстура фона при этом восстанавливается теми

участками, которые выберет пользователь и программа конструирует все после удаления (рисунок 1.8). Программа не устраняет дефекты автоматически, но при этом качество изображения почти не теряется. Процесс детектирования в данном программном обеспечении полуавтоматический. Пользователю необходимо самостоятельно выделять сильно поврежденные участки изображения, что делает программу не очень удобной и выделение дефектных участков занимает большое количество времени.

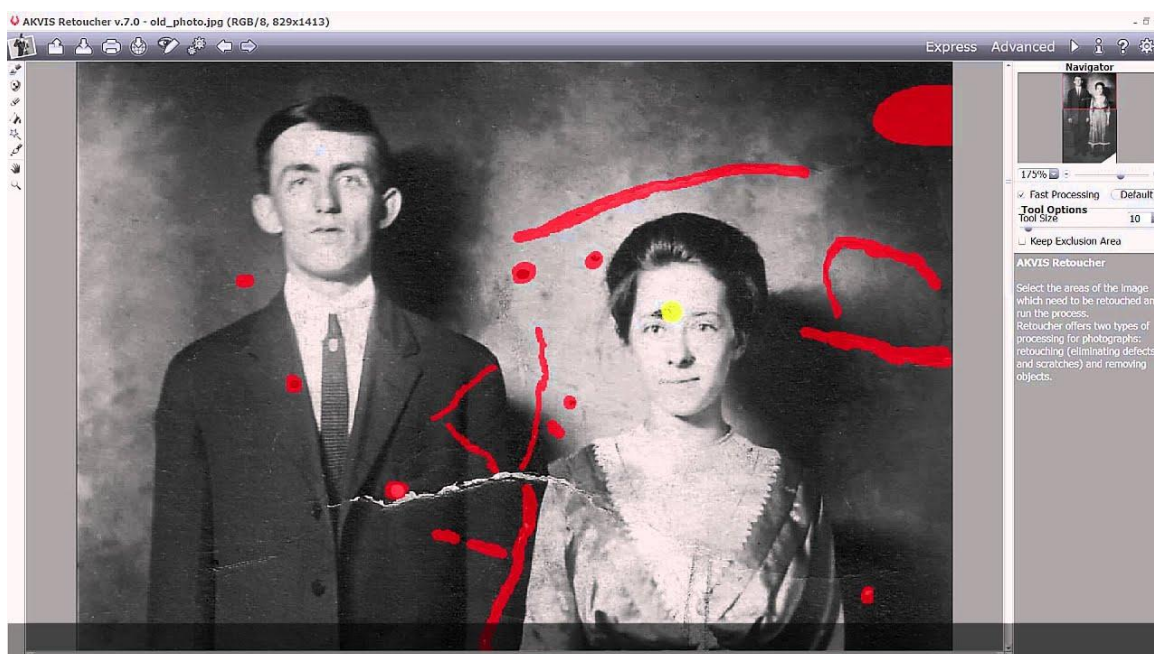


Рисунок 1.8 – AKVIST Retoucher

1.4.2 JPEG Recovery Pro

JPEG Recovery Pro – это программа для восстановления поврежденных изображений и фотографий в формате JPEG.

Большинство подобных проблем JPEG Recovery Pro решает в автоматическом режиме: он может обрабатывать сразу несколько файлов и пользователю не придется добавлять фотографии по одной. Достаточно указать каталог с изображениями и выделить поврежденные, либо выбрать все, после чего указать, куда будут сохраняться восстановленные файлы, и нажать кнопку «Recovery» (рисунок 1.9).

В особо сложных ситуациях автоматика не справляется. Для этого пользователю, как и в прошлом примере, необходимо выделить поврежденные участки. Но в отличие от предыдущего пользователь может использовать встроенный редактор, в котором можно вручную работать с поврежденными изображениями: указывать контрольные точки, удалять или вставлять пиксели между ними, использовать увеличение для пиксельного выделения и так далее.



Рисунок 1.9 – JPEG Recovery Pro

1.5 Постановка задачи

Пусть X – входное изображение с различными повреждениями и импульсными шумами или изображение с низким разрешением. Пусть Y – исходное изображение X , без каких-либо шумов и повреждений с высоким разрешением. Цель данной работы – получить изображение $F(X)$, такое, чтобы оно было максимально схожим с Y . Необходимо разработать алгоритмы и методы для восстановления изображений. В качестве поврежденных изображений будут рассматриваться:

- цветные изображения с различными видами шумов и дефектами;
- сжатые и низкокачественные изображения.

Необходимо разработать и реализовать методы для восстановления изображений, с повреждениями описанные выше.

1.6 Выводы по первому разделу

В данном разделе были выдвинуты основные требования к программе восстановления изображений и постановка задачи. Для детектирования повреждений за основу был взят метод локальных бинарных окрестностей. В разделе рассмотрены наиболее эффективные методы интерполяции для устранения царапин, дефектов и различных шумов на изображениях и их основные недостатки. Так же были рассмотрены сверточные нейронные сети для устранения артефактов сжатия, модели, повышающие качество изображений. Как оказалось, модели не являются универсальными и эффективны только для тех видов повреждений, на которых обучались.

– коэффициент,
и необходимы для нормального распределения яркости.

$$- \text{ — } - \text{ — } \quad (2.3)$$

Таким образом, получаем одноканальное изображение, то есть изображение в градациях серого, что позволяет уменьшить время работы программы и существенно сократить количество вычислений. Дескрипторы локальных бинарных окрестностей при этом не меняются.

LBP оператор позволяет получить матрицу изображения, описывающую текстуру по формуле:

$$(2.4)$$

где – число соседей,

– радиус,

– центральный пиксель,

– окрестность вокруг центрального пикселя,

– функция, которая возвращает 0, если $x > 0$ и 1, если $x < 0$.

В формуле (2.4) последовательно выбирается окрестность 3×3 , отсюда следует, что $P = 8$, так как центральный пиксель не считается. Функция возвращает двоичное число в окрестности P , которое переводится в десятичную систему счисления. Далее выбирается следующий центральный пиксель, и таким образом, формируется матрица локальных бинарных окрестностей.

Для выделения всех переходящих текстур необходимо найти все дескрипторы, у которых переходов с 0 на 1 и наоборот больше двух. Для вычисления значимых дескрипторов используется следующая формула:

$$(2.5)$$

где – число переходов между 1 и 0,

– пороговое значение на количество переходов.

Таким образом, формируется гистограмма для локальной области, представляющая собой дескриптор для анализа локальных областей в изображении на наличие царапин (рисунок 2.1).

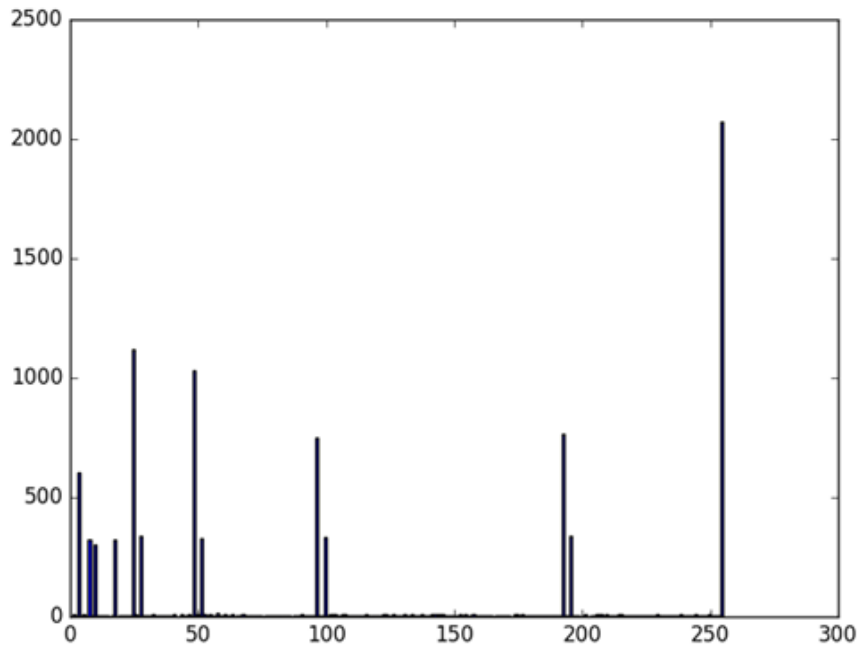


Рисунок 2.1 – Гистограмма для неоднородных окрестностей

2.1.2 Метод опорных векторов

Дескрипторы локальных бинарных окрестностей находят все неоднородные окрестности изображения. Для классификации и соотнесения найденных участков к классу повреждений или к классу других сложных текстур изображения используется метод опорных векторов (Support vector machine, SVM) [15].

Пусть заданы:

- множество X обучающих объектов, в данном случае с царапинами, заданных векторами признаков x_1, x_2, \dots, x_n , (X является подмножеством евклидова пространства размерности d);
- множество Y ответов для обучающих объектов:

Тогда задача классификации неоднородных участков состоит в построении такой функции классификатора, которая каждому вектору сопоставляет правильный ответ. В нашем случае либо -1, либо 1.

Так как задача классификации в данном случае не является линейной, то в методе опорных векторов в качестве функции выбрана плоскость, расстояние до которой ближайших векторов обоих классов равны. При этом для всех объектов одного класса должно выполняться неравенство $\|x - x_i\| \leq \rho$, а для всех объектов другого класса – неравенство $\|x - x_j\| \geq \rho$.

Уравнение разделяющей плоскости имеет следующий вид:

$$(2.6)$$

где n – размерность пространства признаков,
 \vec{n} – направляющий вектор,
 ρ – скалярный порог.

На первом этапе из множества обучающих примеров отбираются опорные векторы, на основе которых строится разделяющая плоскость. Данный этап называется обучающим. Затем идет этап распознавания и заключается в том, что на вход полученного классификатора подается одноканальное изображение, значение пикселей которого вытянуты в вектор. О классовой принадлежности вектора ничего не известно. Классификатор должен дать ответ, к какому классу относится вектор.

В данной работе для построения разделяющей поверхности и классификации используется радиальная базисная функция:

$$(2.7)$$

где \vec{x} – текущий вектор,
 $\vec{x_c}$ – центральный вектор,
 σ – нормализующий параметр.

Пример разделяющей поверхности приведен на рисунке 2.3.

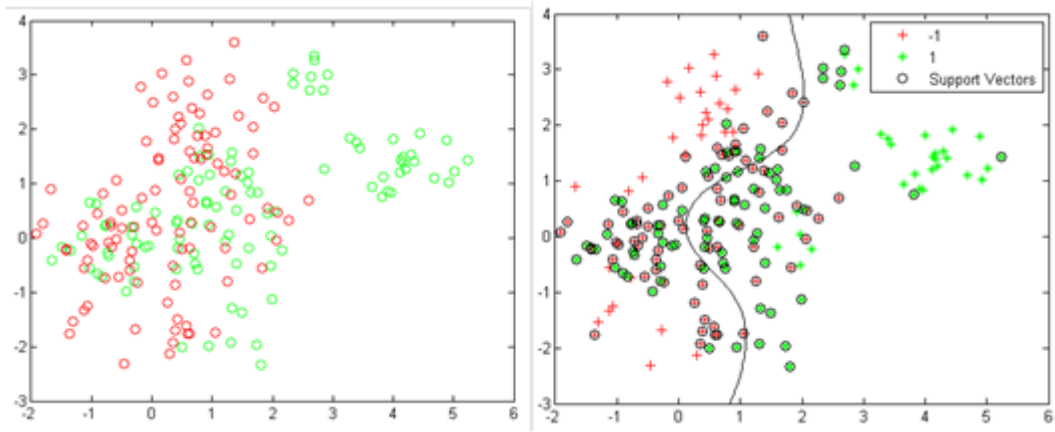


Рисунок 2.2 – Иллюстрация построения разделяющей поверхности

2.2 Метод интерполяции для восстановления изображения

Для восстановления изображений с различными шумами и повреждениями используется метод Телла [17]. Для нахождения недостающих сегментов используется набор уровней и метод быстрого прохода (fast marching method, FMM) [18].

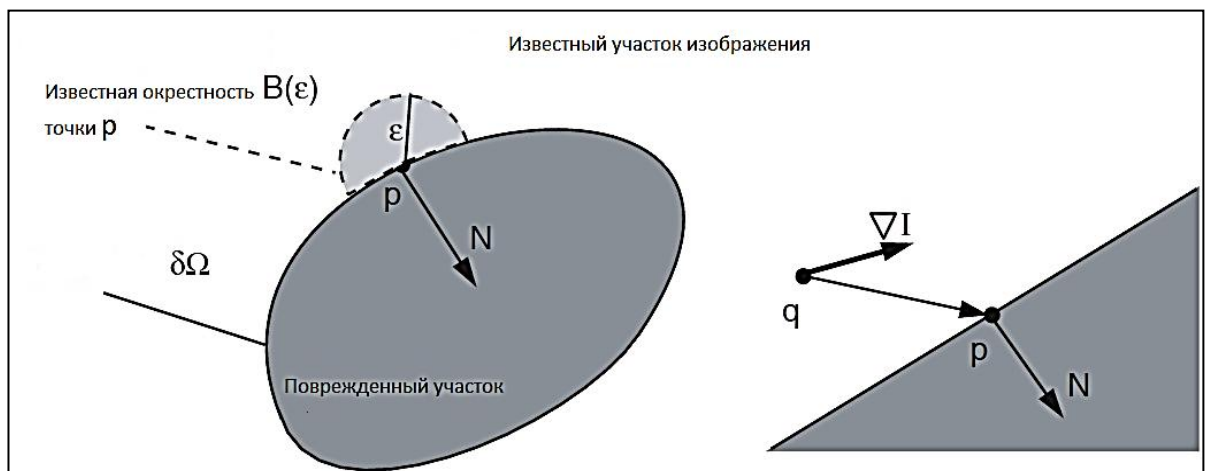


Рисунок 2.3 – Метод Телла для закрашивания поврежденных участков

Используя бинарную маску царапин, построенную с помощью предыдущих алгоритмов (см. п. 2.1), уменьшаем границу с постоянной единичной скоростью и закрашиваем по формуле:

$$(2.8)$$

где ρ – нормализованная весовая функция,

$B(\epsilon)$ – окрестность точки p ,

– заданное изображение в точке .

Формула (2.8) используется для заполнения границы , для остальной части итерационно смещаем границу и заполняем поврежденный участок до тех пор, пока он не закончится. В формуле (2.8) q – сумма пикселей в окрестности , деленное на их количество. Далее берем среднее между q и p и получено значение – это значение пикселя на границе поврежденного участка. Таким образом, вдоль границы закрашиваются все пиксели. Затем граница смещается и заполняется новая граница.

2.3 Модели сверточной нейронной сети для восстановления сжатых изображений

Для улучшения качества изображений с низким разрешением будем использовать сверточную нейронную сеть на основе моделей ARCNN [13, 14] и SRCNN [12]. Пусть X – входное изображение с артефактами сжатия. Y – изображение с высоким качеством, с которым будем сравнивать X . В ходе работы сверточной нейронной сети необходимо получить изображение $F(X)$, такое чтобы оно было максимально схожим с Y . Данная модель состоит из четырех сверточных слоев (рисунок 2.4):

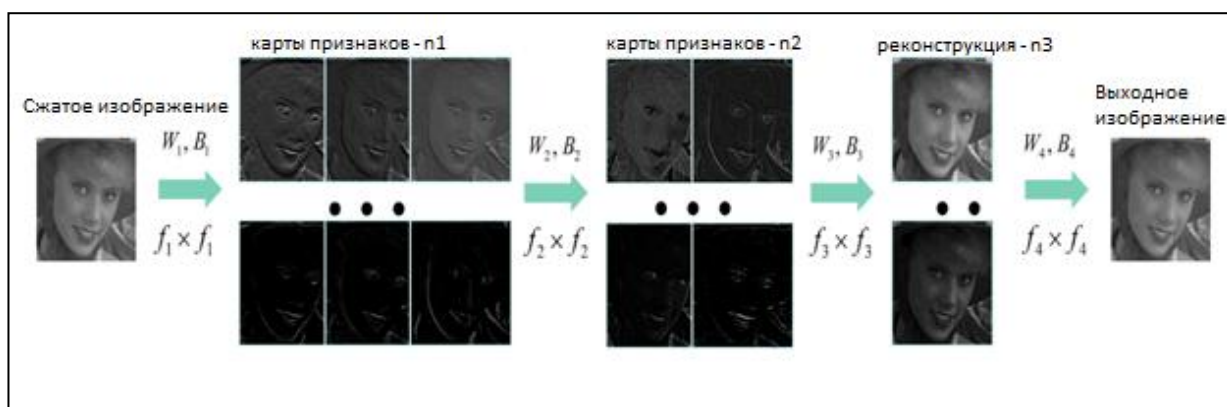


Рисунок 2.4 – Модель сверточной нейронной сети

Первый слой – получение патчей. Эта операция позволяет получить отдельные участки изображения, состоящие из нескольких пикселей из изображения низкого разрешения , а затем представить каждый патч в виде

многомерного вектора. Эти векторы содержат набор карт признаков, размер которых равен размерности вектора (n_I карты признаков на рисунке 2.4).

Формулу слоя для получения патчей обозначим :

$$Y = W * X + b$$
 (2.9)

где W – фильтры,
 b – веса,
 $*$ – операция свертки.

Фильтры можно обозначить через формулу:

$$W = \frac{1}{c} * Y$$
 (2.10)

где c – количество каналов изображения Y ,
 W – пространственный размер фильтра, на первом слое он равен 9.

Фильтры осуществляют n_1 операций свертки изображения, на первом слое $n_1 = 64$, и каждая свертка имеет ядро свертки размера .
 Выходные данные слоя содержат n_1 карт признаков.

Веса B_1 представляют собой n_1 – мерный вектор, каждый элемент которого сопоставлен с элементом фильтра W_1 .

В качестве активационной функции первого слоя используется ReLU (Rectified Linear Unit). Формула функции активации выглядит так:

$$f(x) = \max(0, x)$$
 (2.11)

где x – входной сигнал.

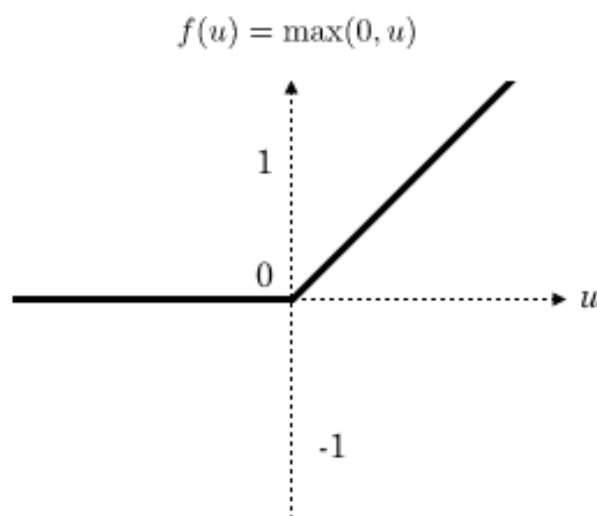


Рисунок 2.5 – Функция активации

Второй слой позволяет улучшить признаки, извлекаемые из изображения. Данный слой используется в модели AR CNN [12]. Второй слой свертки можно описать следующим образом:

$$, \quad (2.12)$$

где \dots – фильтры второго слоя,
 \dots – веса второго слоя.

В качестве активационной функции, как и у первого слоя, используется ReLU.

Третий слой осуществляет операцию нелинейного отображения n_2 – мерных векторов на n_2 – мерные векторы, где $n_2 = 32$, $n_3 = 16$. Эта операция работает только для фильтров с размером 1×1 . Операцию, осуществляемую третьим сверточным слоем можно выразить через формулу:

$$, \quad (2.13)$$

где параметры \dots и \dots аналогичны схожим параметрам для \dots и \dots с той разницей, что теперь фильтры осуществляют n_3 операций, а веса представляют собой n_3 – мерный вектор. Функция активации на данном слое ReLU.

Четвертый слой осуществляет реконструкцию изображения \dots , выходной слой которого изображение (\dots) с высоким разрешением, максимально близкое к эталону. Данный слой можно выразить формулой:

$$, \quad (2.14)$$

где \dots – это с фильтров размерностью \dots , $\dots = 1$, $\dots = 5$,
 \dots – это с-мерный вектор.

Функция активации выходного слоя является линейная функция:

$$, \quad (2.15)$$

где x – входной сигнал.

В модели используется стандартная функция потерь – среднеквадратическая ошибка (Mean Squared Error, MSE):

$$- \dots, \quad (2.16)$$

где $\{w_1, w_2, \dots, w_n\}$ – это параметры нейронной сети, n – количество образцов в обучающей выборке.

В качестве метода минимизации ошибки используется метод градиентного спуска. Суть данного метода сводится к поиску минимума (или максимума) функции за счет движения вдоль вектора градиента. Для поиска минимума движение должно осуществляться в направлении антиградиента. Метод градиентного спуска в соответствии с рисунком 2.10:

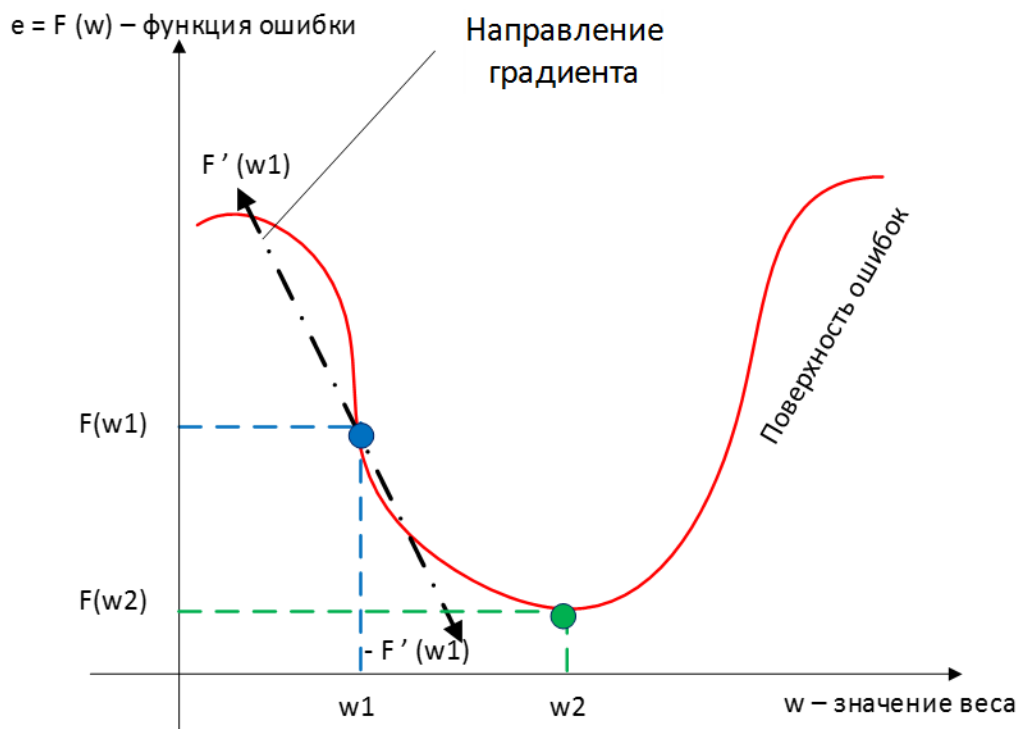


Рисунок 2.6 – Метод градиентного спуска

Градиент функции потерь представляет собой вектор частных производных, вычисляющийся по формуле:

$$\frac{\partial F}{\partial w} = \left[\frac{\partial F}{\partial w_1}, \frac{\partial F}{\partial w_2}, \dots, \frac{\partial F}{\partial w_n} \right], \quad (2.17)$$

где $\frac{\partial F}{\partial w}$ – градиент функции потерь от матрицы весов,

$\frac{\partial F}{\partial w_i}$ – частная производная функции ошибки по весу нейрона,

n – общее количество весов сети.

Производную функции ошибки по конкретному образцу можно записать таким образом:

где $\frac{\partial e_j}{\partial w_{ij}}$ – значение производной функции ошибки по весу w_{ij} между нейронами,

e_j – ошибка нейрона j ,

$\frac{\partial \sigma_j}{\partial z_j}$ – значение производной функции активации по ее аргументу для нейрона j ,

z_j – выход нейрона предыдущего слоя.

2.4 Выводы по второму разделу

Для детектирования был выбран оператор локальных бинарных окрестностей, так как данный оператор хорошо работает с текстурами. В данной главе описана математическая модель оператора. Для классификации найденных оператором локальных бинарных окрестностей текстур используется метод опорных векторов, который основан на обучении и для классификации использует опорные векторы, метод также описан в данном разделе. Найденные повреждения устраняются методом Телеа [18]. Выбранный метод интерполяции является лучшим среди всех рассмотренных методов. Он не заполняет поврежденный участок одним цветом и не использует блочный алгоритм, а обрабатывает только границы поврежденного участка. Метод Телеа гарантирует сохранения границ в изображении. Математическая модель метода описана в данной главе. Для восстановления сжатых изображений подробно описана модель сверточной нейронной сети. Сверточная нейронная сеть разработана на основе моделей SR CNN и AR CNN, содержит 4 сверточных слоя и является более универсальной.

3 РАЗРАБОТКА АЛГОРИТМОВ ВОССТАНОВЛЕНИЯ ИЗОБРАЖЕНИЙ

3.1 Алгоритм предварительной обработки изображения

Перед использованием изображения для восстановления и обработки его необходимо подготовить. Схему алгоритма обработки изображения можно описать следующим образом:

Для работы с изображением его необходимо преобразовать в матричный вид. Входное изображение имеет три канала (красный, синий, зеленый), для того чтобы не использовать матрицы для всех трех каналов целесообразнее будет преобразовать его в одноканальное изображение.



Рисунок 3.1 – Алгоритм предварительной обработки

3.2 Алгоритм LBP

Каждый пиксель изображения представляется в виде числа, в случае черно-белого изображения, и тройкой чисел, в случае цветного изображения. Но информации становится слишком много даже для изображения в градациях серого. Для классификации текстур редко используется все пиксели изображения, достаточно только основных пикселей, описывающих текстуру изображения. Оператор локальных

бинарных окрестностей (LBP) применяется в задаче классификации текстур. Данный оператор представляет каждый пиксель изображения в виде бинарного числа и является очень эффективным за счет того, что работает только с целочисленной арифметикой.

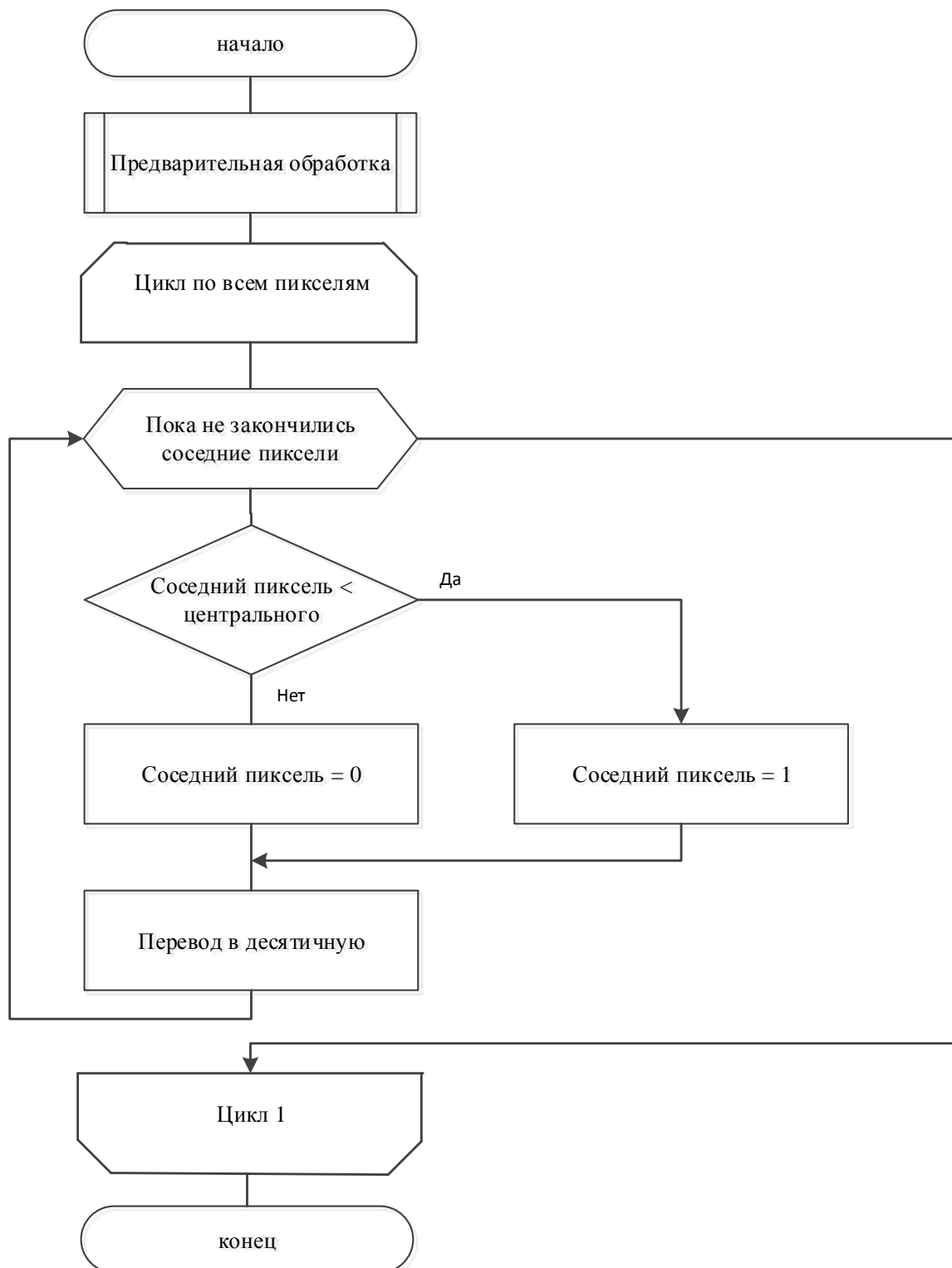


Рисунок 3.2 – Алгоритм LBP

LBP оператор устойчив к изменениям яркости и освещения, так как описывает только структуру изображения, и каждый пиксель зависит от соседних пикселей. Он принимает центральный пиксель в качестве порогового значения и сравнивает значение яркости с соседними пикселями. Если это значение больше центрального, то пиксель принимает значение 1. Если же меньше – 0. Полученные значения вокруг центрального пикселя это 8-битное двоичное число, которое переводится в десятичную систему счисления. Всего вариантов таких чисел $2^8 = 256$. Таким образом, каждому пикселю изображения присваивается одна из 256 меток, характеризующих его. Для наглядности все значения можно собрать в гистограмму и визуально увидеть есть ли повреждения на изображении.

3.3 Алгоритм создания бинарной маски структуры изображения

Царапины на изображении, как уже говорилось в первой главе, это линии, нарушающие однородность фона. Оператор LBP позволяет получить характеристику структуры изображения. Для того что бы найти царапины и другие шумы на изображении, необходимо извлечь все неоднородные участки изображения. Как правило, однородные участки изображения, полученные из LBP оператора, в большинстве случаев это те, которые близки к 256 или к 0. Например «11111111» или «00000000» являются на изображении пятном или частью какого-то объекта, а число «00111111» описывает границу. Во второй главе описывались значимые окрестности, это те, у которых количество переходов больше двух, например, «01100111». Для создания бинарной маски нужно найти все участки изображения, нарушающие однородность. Изначально всем значениям бинарной маски присваиваем значение 0. Так как границы изображения известны, остается найти все небольшие по отношению к изображению объекты, которые нарушают структуру и однородность фона и присвоить им значение 1.

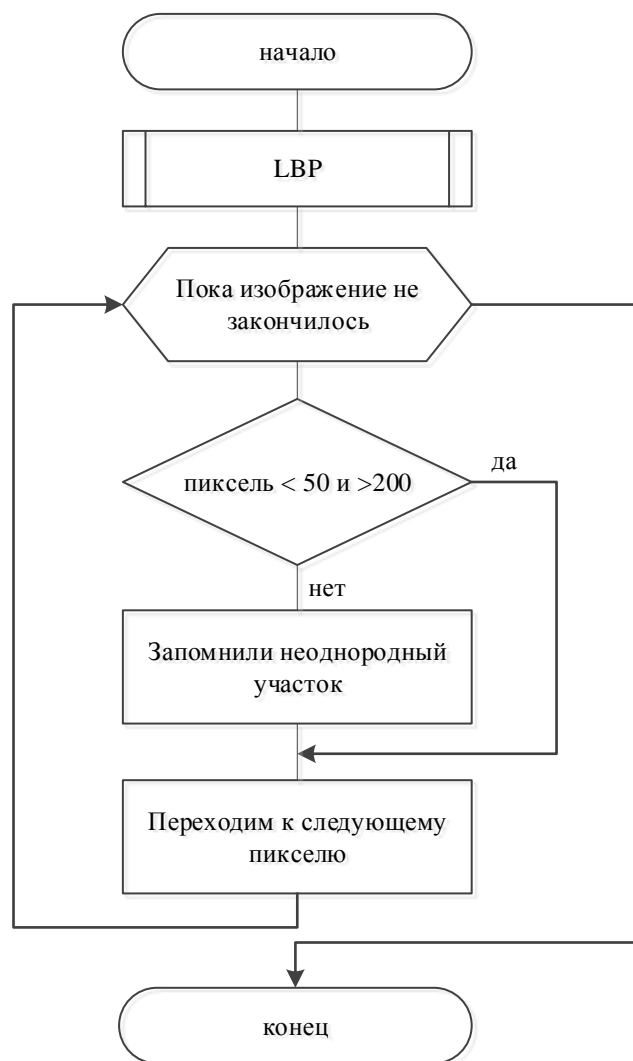


Рисунок 3.3 – Алгоритм создания бинарной маски

3.4 Алгоритм классификации

Для классификации текстур в полученной бинарной маске используется метод опорных векторов (SVM). Из алгоритма создания бинарной маски все однородные пиксели изображения равны 0, а неоднородные равны 1. SVM обучается изображениями 20×20 , которые были предварительно обработаны. Объем обучающей выборки 100 изображений, 50 изображений с царапинами и ответами 1, и 50 изображения со сложными участками и границами с ответами 0. Алгоритм классифицирует оставшиеся участки и создается итоговая бинарная маска царапин.

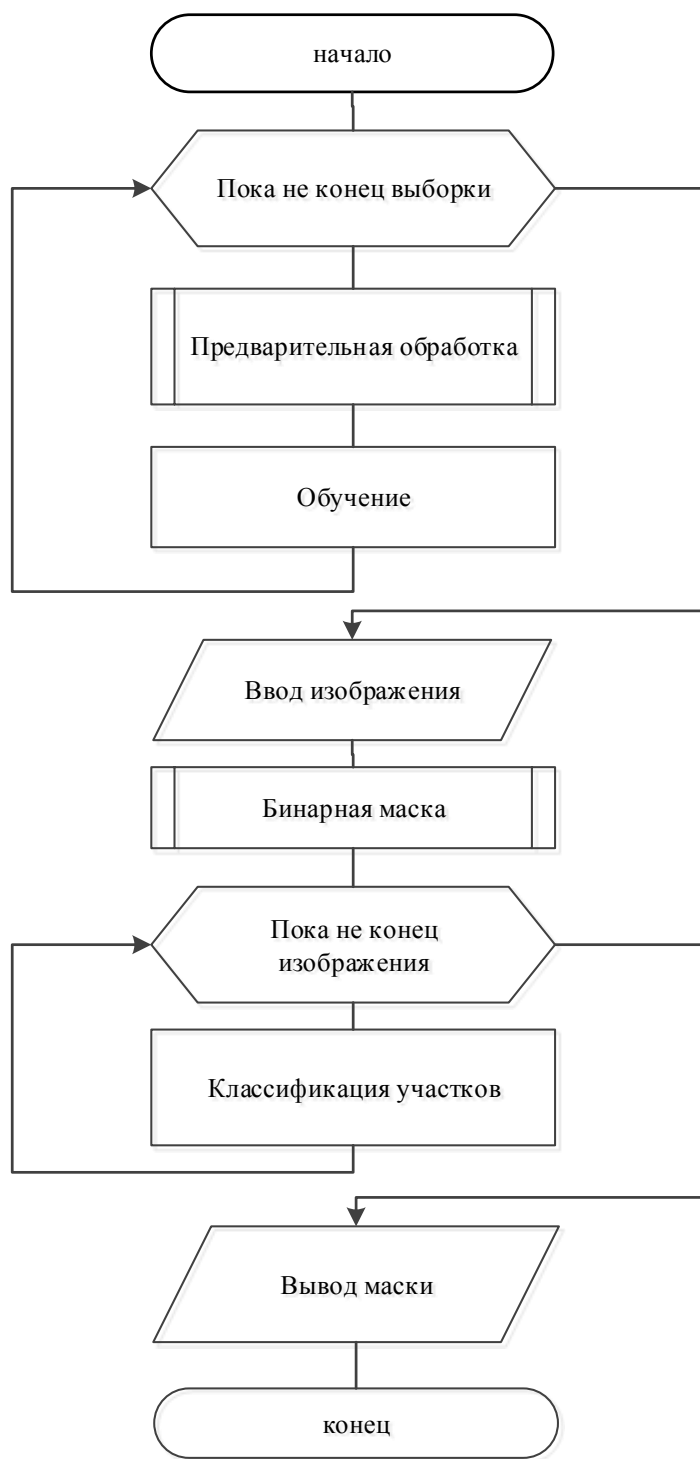


Рисунок 3.4 – Алгоритм SVM

3.5 Алгоритм восстановления изображения

Восстановление изображения осуществляется методом интерполяции Телеа. Данный метод находит границы с помощью бинарной маски и заполняет ближайшие к границе пиксели. Значение пикселя вычисляется как

среднее взвешенное в известной окрестности изображения рядом с границей. Далее итерационно смещая границу, заполняются все пиксели на этой границе.



Рисунок 3.5 – Алгоритм Телеа для восстановления изображений

3.6 Структура сверточной нейронной сети

Обучение сети осуществлялось на наборе, который содержит 400 изображений. В обучающей выборке изображения сжимались в 4 раза методом интерполяции, для этого была использована библиотека на Python `scipy.ndimage`. Данная библиотека позволяет сжимать изображения и возвращать их до исходных размеров, но при этом теряется качество. Поскольку модель нейронной сети состоит только из сверточных слоев, то можно вычислить количество параметров. Число обучаемых параметров равно 106561. Структура сверточной нейронной сети представлена в таблице 3.1.

Таблица 3.1 – Структура нейронной сети

Слои (Тип)	Выходная форма	Число параметров
Входное изображение	(256,256,1)	0
Первый сверточный слой (Conv2D)	(256,256,64)	5248
Функция активации (Activation)	(256,256,64)	0
Второй сверточный слой (Conv2D)	(256,256,32)	100384
Функция активации (Activation)	(256,256,32)	0
Третий сверточный слой (Conv2D)	(256,256,16)	528
Функция активации (Activation)	(256,256,16)	0
Четвертый сверточный слой (Conv2D)	(256,256,1)	401
Функция активации (Activation)	(256,256,1)	0

В качестве оптимизатора был использован алгоритм «Адам». Темп обучения 0,0003. Модель была разработана с использованием библиотеки `keras`. Данная библиотека нацелена на работу с сетями глубокого обучения, при этом является очень компактной. Сверточная нейронная сеть была реализована и обучена на языке Python 3.6.

3.7 Разработка модуля восстановления изображений

3.7.1 Диаграмма компонентов

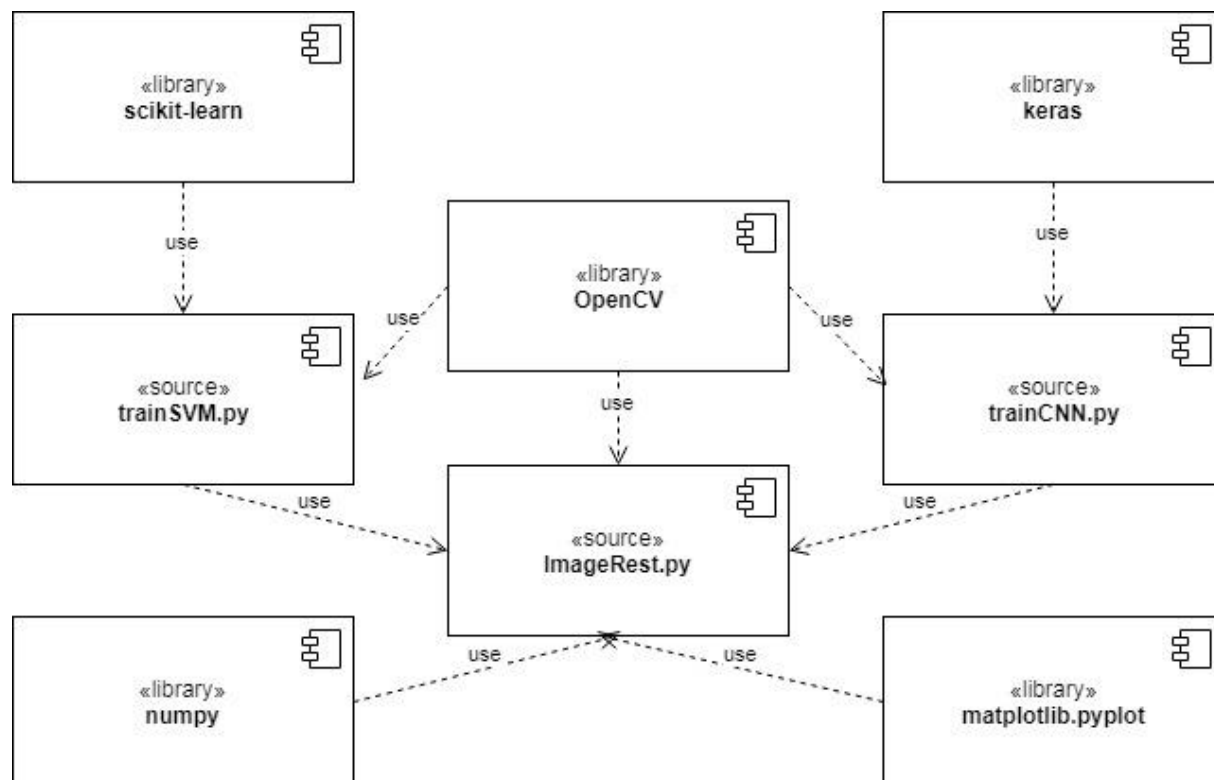


Рисунок 3.6 – Диаграмма компонентов модуля

3.7.2 Описание библиотек модуля

Модуль `trainSVM.py` обучает классификатор для определения текстур изображения, найденные LBP оператором. Для обучения используется библиотека `scikit-learn`. Данная библиотека предоставляет готовые функции для обучения классификатора и намного упрощает этап обучения и предсказания данных. Так же модуль использует библиотеку `OpenCV` для загрузки изображений и их предварительной обработки.

В модуле `trainCNN.py` содержится модель нейронной сети и обучение. Данный модуль использует библиотеку `keras`. `Keras` предоставляет готовые сверточные слои. Так же библиотека содержит готовые функции активации, различные оптимизаторы и функции потерь. Сохранение и загрузку обученных весов так же позволяет сделать данная библиотека.

Основной модуль ImageRest.py аналогично использует библиотеку OpenCV для загрузки и сохранения изображений. Кроме того, библиотека используется для закрашивания поврежденных пикселей методом Телеа. Предварительно обработанные изображения в серых тонах являются матрицами. Для удобства работы с матрицами используется библиотека numpy. Для визуализации работы LBP необходима библиотека matplotlib.pyplot. Библиотека дает возможность построить гистограмму LBP оператора, так же она необходима для построения и вывода графиков.

3.8 Выводы по разделу

В данном разделе были представлены схемы и описание алгоритмов для нахождения сложных текстур, их классификация методом опорных векторов на поврежденные и не поврежденные участки. Был разработан алгоритм создания бинарной маски. Бинарная маска необходима для заполнения поврежденных участков. Закрашивание поврежденных участков выполнялось методом Телеа, алгоритм которого представлен в данной главе. Также в данной главе подробно описана структура сверточной нейронной сети, которая используется для восстановления сжатых и низкокачественных изображений. Приведено описание используемых библиотек и диаграмма компонентов. Все алгоритмы были реализованы на языке Python 3.6. Для реализации разработанной модели сверточной нейронной сети использовалась библиотека глубокого обучения.

4 ИССЛЕДОВАНИЕ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

4.1 Метрики оценки эффективности алгоритмов

Для оценки результатов алгоритмов и модели нейронной сети были реализованы общепринятые метрики сходства изображений:

- 1) метод среднеквадратической ошибки (MSE);
- 2) пиковое отношение сигнала к шуму (PSNR);
- 3) индекс структурного сходства (SSIM).

Среднеквадратическая ошибка двух изображений является самой простой метрикой и вычисляется по формуле:

$$\text{MSE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (I(i,j) - J(i,j))^2, \quad (4.1)$$

где $I(i,j)$ – значение канала пикселя с координатами (i,j) эталонного изображения,

$J(i,j)$ – значение канала пикселя с координатами (i,j) сравниваемого изображения,

M – ширина изображения,

N – высота изображения.

Метрика пикового отношения сигнала к шуму (PSNR) характеризует соотношение между максимумом возможного значения сигнала и мощностью шума, искажающего значения сигнала. Данная метрика обычно измеряется в логарифмической шкале в децибелах. PSNR определяется через среднеквадратичное отклонение (MSE) и вычисляется по следующей формуле:

$$\text{PSNR} = 10 \log_{10} \left(\frac{I_{\max}^2}{\text{MSE}} \right), \quad (4.2)$$

где I_{\max} – это максимальное значение пикселя изображения.

В отличие от предыдущих метрик, метрика SSIM является самой точной и учитывает искажения яркости и контрастности пикселей. Данный метод не привязан к специфике изображения и искажениям, а основывается

на статистическом анализе отдельных блоков входного изображения и дальнейшем сравнении полученных результатов со значениями эталонного изображения. Значение метрики вычисляется следующим образом:

(4.3)

где μ – среднее арифметическое значений каналов пикселей изображения ,
 σ – стандартное отклонение значений каналов пикселей изображения ,
 σ_{12} – ковариация значений каналов пикселей изображений I_1 и I_2 ,
 c – выравнивающий коэффициент.

Нахождение SSIM для всего изображения сводится к следующему: «скользящее» окно размером $A \times B$, начиная из верхнего левого угла, проходит пиксель за пикселем по всем строкам и столбцам этого изображения, при этом на каждом шаге вычисляется SSIM для текущей области. Выходное значение метрики от 0 до 1. Чем ближе к 1, тем больше сходство изображений и наоборот, чем ближе значение к 0, тем меньше сходство между изображениями.

4.2 Оценка результатов

4.2.1 Результат работы алгоритма восстановления изображения

Для оценки результатов и сравнения основного метода Телера и доработанного были использованы выше названные метрики и несколько изображений с разным процентом повреждений (15%, 8%, 5%).



Рисунок 4.1 – Примеры поврежденных изображений



Рисунок 4.2 – Примеры восстановленных изображений

Поврежденные изображения сравнивали с эталонными и получили значения для сравнения. После восстановления изображений снова сравнили их с эталонными и проанализировали результаты работы алгоритма, на сколько изменились значения метрик. В таблицах 4.1–4.3 приведены сравнения основного алгоритма Телеа, улучшенного и все значения метрик.

Таблица 4.1 – Оценка алгоритма с помощью метрики MSE

Номер изображения	Поврежденное изображение	Метод Телеа	Улучшенный метод
1	7,07	6,79	6,77
2	6,05	4,45	4,43
3	1,07	0,87	0,87

Таблица 4.2 – Оценка алгоритма с помощью метрики PSNR

Номер изображения	Поврежденное изображение	Метод Телеа	Улучшенный метод
1	39,56	39,71	39,78
2	40,30	51,28	51,31
3	47,82	59,93	59,93

Как видно из полученных значений, алгоритм отлично восстанавливает изображения с небольшим процентом повреждений, человеку визуально трудно заметить разницу. Улучшенный алгоритм Телеа в сравнении с основным лучше восстанавливает изображения с большим количеством

текстур, границ, но с большими повреждениями результат практически одинаковый.

Таблица 4.3 – Оценка алгоритмов с помощью метрики SSIM

Номер изображения	Поврежденное изображение	Метод Телеа	Улучшенный метод
1	0,70	0,82	0,79
2	0,81	0,93	0,91
3	0,87	0,96	0,96

4.2.2 Результат работы сверточной нейронной сети

Для оценки и сравнения результатов сверточной нейронной сети были использованы изображения с разной степенью сжатия (Рисунок 4.3).

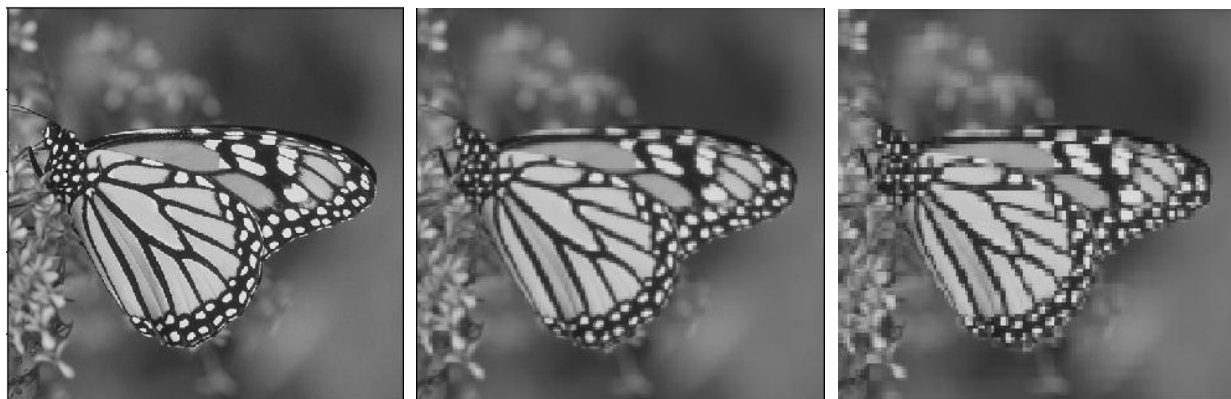


Рисунок 4.3 – Пример сжатых изображений (слева оригинальное, по центру сжатое в два раза, справа сжатое в 4 раза)



Рисунок 4.4 – Пример восстановленных изображений (слева оригинальное, по центру сжатое в два раза, справа сжатое в 4 раза)

Так как визуально сложно заметить разницу в результатах работы модели SR CNN и модели разработанной в данной работе, сравнения

производились по метрикам качества изображений. Изображение 1 было сжато в 2 раза, изображение 2 было сжато в 4 раза. Данные представлены в таблицах 4.4–4.6.

Таблица 4.4 – Оценка СНН с помощью метрики MSE

Номер изображения	Сжатое изображение	Восстановленное изображение	Модель SRCNN
1	7,07	6,79	6,97
2	6,05	4,45	5,12

Таблица 4.5 – Оценка СНН с помощью метрики PSNR

Номер изображения	Сжатое изображение	Восстановленное изображение	Модель SRCNN
1	31,15	37,49	37,53
2	20,04	28,45	25,34

Таблица 4.6 – Оценка СНН с помощью метрики SSIM

Номер изображения	Сжатое изображение	Восстановленное изображение	Модель SRCNN
1	0,69	0,82	0,83
2	0,53	0,71	0,67

4.2.3 Результаты обучения

Модели обучались на выборке из 400 изображений, сжатых в 4 раза. Все рассмотренные модели и модель, разработанная в данной работе, обучались почти одинаковое количество времени, так как разница сверточных слоев небольшая, за исключением модели супер-разрешения. На рисунке 4.5 представлен график обучения сверточной нейронной сети, которая была разработана на основе моделей супер-разрешения и устранения артефактов сжатия. Как видно из графика, модель практически полностью обучилась уже на шестидесятой эпохе, но при этом сохраняла положительную динамику обучения.

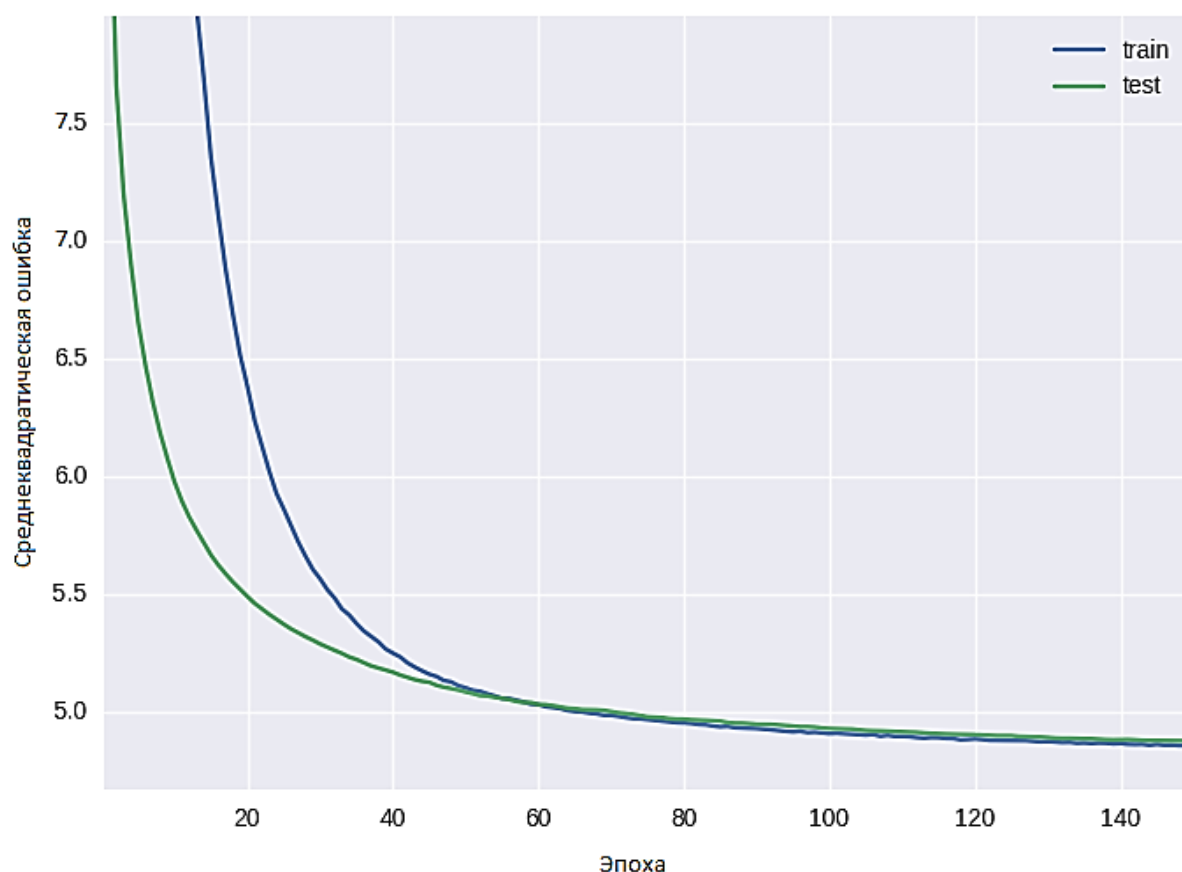


Рисунок 4.5 – График обучения сверточной нейронной сети

Обучение и реализация сверточной нейронной сети производилась на языке Python 3.6 с использованием библиотеки keras. Данная библиотека позволяет ускорять процесс обучения и содержит готовые функции потерь. Стоит отметить, что модели сверточных нейронных сетей показывали разные результаты при одном объеме обучающей выборки, то есть, качества восстановления изображения в большей степени зависит от структуры сверточной нейронной сети, а не от обучающей выборки и типа изображений в обучающей выборке.

4.3 Выводы по разделу

В данном разделе были выбраны метрики качества. По данным метрикам провели анализ результатов работы. Оценили качество работы алгоритма восстановления поврежденных изображений методом Телеа и доработанным нами методом. В изображениях с большим количеством

повреждений улучшенный метод показывал результаты лучше основного, при небольших повреждениях (менее 10% повреждений) результаты были одинаковы. Так же в данной работе проведены сравнения моделей сверточных нейронных сетей. Модель, разработанная в данной работе, показала результаты намного лучше модели SR CNN с изображениями, сжатыми в 4 раза. При небольшой степени сжатия (в 2 раза) модели примерно равны. Сравнили с помощью метрик качества, восстановленные и поврежденные изображения.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы были разработаны алгоритмы для детектирования поврежденных участков и их особенности, методы интерполяции для восстановления поврежденных участков. Среди методов интерполяции в работе был выбран метод Телеа, который закрашивает поврежденные пиксели согласно известной границе. Такой метод закрашивания является наилучшим среди всех существующих методов. В работе был реализован и улучшен метод интерполяции. Как показали экспериментальные данные, метод начал работать лучше с большим количеством повреждений. Импульсные шумы стали обрабатываться лучше.

Для выделения сложных структур использовался оператор локальных бинарных окрестностей (LBP). LBP является мощным средством для работы с изображениями. С помощью данного алгоритма были выделены все неоднородные структуры. Оператор так же может применяться для распознавания объектов. Как показывают современные программные реализации, данные полученные LBP отправляют на вход нейронной сети. В данной же работе, результаты, полученные LBP, сложные текстуры изображения, идут на вход классификатора.

В качестве классификатора использовался метод опорных векторов (SVM). Данный метод исполняет роль бинарного классификатора. SVM делит поврежденные участки и просто неоднородные текстуры изображения на два класса. Для всех вышеописанных алгоритмов и методов было приведено описание во второй и третьей главе.

В рамках данного исследования была разработана сверточная нейронная сеть. Она используется для обработки данных, имеющих упорядоченную, сеточную структуру. Сверточную нейронную сеть применяют для решения множества задач в области обработки изображений и в других областях, что делает их одной из самых используемых архитектур нейронных сетей. Для эффективного решения восстановления сжатых

изображений в работе были изучены лучшие модели сверточных нейронных сетей. К ним относится сеть ARCNN, которая является продолжением модели SRCNN, решающей проблему супер-разрешения, путём добавления слоя для улучшения, извлекаемых из изображения признаков. На их основе в работе была предложена другая модель сверточных нейронных сетей, которая на экспериментальных данных показала результаты лучше, чем ARCNN, но время работы немного увеличилось, доли секунды не существенны в данном случае. Так как данные алгоритмы не работают в режиме реального времени, акцент был сделан на качество и универсальность обработки. Так же по экспериментальным данным был сделан вывод, что качество восстановления в большей степени зависит не от обучающей выборки и ее размера, а в большей степени от модели сверточной нейронной сети.

Модель нейронной сети была разработана и обучена на языке Python с использованием библиотеки keras. Keras предоставляет готовые слои, что упрощает и ускоряет время работы сверточной нейронной сети для исследований. Как показали результаты эксперимента, разработанная сеть, является более стабильной и может одинаково хорошо восстанавливать как сильно сжатые изображения, так и слабо сжатые.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Воронин, В.В. Алгоритм реконструкции изображений на основе анализа локальных бинарных окрестностей / В.В. Воронин. – Москва: Издательство радиотехника, 2011. – 84 с.
- 2 Гонсалес, Р.Р. Цифровая обработка изображений / Р.Р. Гонсалес. – Москва: Техносфера, 2012. – 489 с.
- 3 Воронин, В.В. Автоматизированное детектирование дефектов при реконструкции архивных фотодокументов / В.В. Воронин. – Научно-технические ведомости СПбГПУ. – Санкт Петербург, 2012. – С. 50–54.
- 4 Воронин, В.В. Использование метода реконструкции двумерных сигналов при восстановлении архивных фотографий / В.И. Марчук, Н.В. Гапон, Р.А. Сизякин, В.В. Воронин. – Успехи современной радиоэлектроники. – Москва: Издательство Радиотехника, 2011. – С. 16–19.
- 5 Barni, M. Image processing for virtual restoration of artworks / M. Barni, F. Bartolini, V. Cappellini // – IEEE Multimedia, 2000. – Pp. 34–37.
- 6 Qingquan, L.I. Novel Approach to Pavement Image Segmentation Based on Neighboring Difference Histogram Method. Image and Signal Processing / L.I. Qingquan, L.U. Xianglong. – CISP '08. Congress on, 2008. – Pp. 792–796.
- 7 Ojala, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns / T. Ojala, M. Pietikainen, M. Maenpaa. – IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. Pp. 971–987.
- 8 Msjukov, A. Multiscale modification of Shepard's method for multivariate interpolation of scattered data. / A. Masjukov, V. Masjukov. – Mathematical modeling and analysis, 2005. Pp 467–472.
- 9 Xiang, C. An improved exemplar based image inpainting algorithm / C. Xiang, Duan, Y. Cao. – Canada: International Conference on Computer Science & Education, 2014. Pp. 770–775.

10 Zhai, D.H. Image inpainting algorithm based on partition block of damaged region / D.H. Zhai, J.Yu, W. X. Duan, J. Xiao. – Journal of Image and Graphics, 2014. Pp. 835–842.

11 Дорогой, Я.Ю. Архитектура обобщенных сверточных нейронных сетей / Я.Ю. Дорогой. – Вестник Национального технического университета Украины. – 2011. С. 229–234.

12 Dong, C. Convolution Networks for Compression Super Resolution / C. Dong, D. Yubin, C. Loy. – Xiaoou Tang – The Chinese University of Hong Kong, 2015. Pp. 71–82.

13 Dong, C. Compression Artifacts Reduction by a Deep Convolutional Network, in Proceedings of International Conference on Computer Vision (ICCV) / C. Dong, D. Yubin, L. Chen. – Xiaoou Tang, – 2015. Pp. 33–47.

14 Dong C. Compression Artifacts Reduction by a Deep Convolutional Network / C. Dong, Y. Deng, C. Change Loy. – Xiaoou Tang, The Chinese University of Hong Kong, 2015. Pp. 61–79.

15 Lin, J. Potholes Detection Based on SVM in the Pavement Distress Image, Ninth International Symposium on Distributed Computing and Applications to Business / J. Lin, L. Yayu. – Engineering and Science. – 2010. Pp. 544–547.

16 Barni, M. Image processing for virtual restoration of artworks / M. Barni. – IEEE Multimedia, 2000. – Pp. 34–37.

17 Telea, A. Alexandru Telea an Image Inpainting Technique Based on the Fast Marching Method / A. Telea. – IEEE Multimedia, 2004. – Pp. 417–436.

18 Garrido, S. Voronoi diagram and fast marching applied to path planning / in Robotics and Automation / S. Garrido. – IEEE International Conference on, may 2006. Pp. 3049–3054.

ПРИЛОЖЕНИЕ 1

Исходный код программы