**Федеральное государственное автономное образовательное учреждение**

**Высшего образования   
«Национальный исследовательский университет   
«Высшая школа экономики»**

**Московский институт электроники и математики им. А.Н.Тихонова**

Направление подготовки

**«10.03.01 Информационная безопасность»**

Образовательная программа **«Информационная безопасность»**

**О Т Ч Е Т**

**о прохождении**

**учебной практики**

**Студент** Александров М.С.БИБ212⠀

(Фамилия И.О.) номер группы

**Руководитель практики студента:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Младший научный сотрудникинформационной безопасности киберфизических систем департамента электронной инженерии МИЭМ НИУ ВШЭ |  | Мельман А.С. |  |  |
| должность и место работы |  | Фамилия И.О. |  | подпись |

**Руководитель практики от НИУ ВШЭ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Зав. каф. информационной безопасности киберфизических систем департамента электронной инженерии МИЭМ НИУ ВШЭ |  | Евсютин О.О. |  |  |
| должность и место работы |  | Фамилия И.О. |  | подпись |

**Практика пройдена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва, 2023**

**СОДЕРЖАНИЕ**

[1 Введение 3](#_Toc130468950)

[2 Краткая характеристика организации 4](#_Toc130468951)

[3 Описание профессиональных задач 5](#_Toc130468952)

[4 Исполненное индивидуальное задание 6](#_Toc130468953)

[4.1 Построение программы на языке Python 7](#_Toc130468954)

[4.2 Эксперименты и подсчет метрик 10](#_Toc130468955)

[5 Заключение 19](#_Toc130468956)

[6 Список использованных источников 20](#_Toc130468957)

# 1 Введение

Целью прохождения практики является закрепление, расширение и углубление полученных теоретических знаний и приобретение первоначальных практических навыков в решении конкретных проблем.

Задачи практики:

- закрепление и углубление теоретических знаний по прослушанным за время обучения в университете дисциплинам;

- формирование и совершенствование базовых профессиональных навыков и умений в области информационной безопасности;

- знакомство и отработка навыков работы с реальными исследовательскими, промышленными и образовательными проектами;

- формирование информационной компетентности с целью успешной работы в профессиональной деятельности;

- получение навыков самостоятельной работы, а также работы в составе коллектива;

- обработка полученных материалов и оформление отчета о прохождении практики.

# 2 Краткая характеристика организации

МИЭМ НИУ ВШЭ готовит специалистов, бакалавров и магистров в области электроники, информационных технологий, телекоммуникаций, вычислительной техники, прикладной математики, кибернетики и дизайна.

Педагогический коллектив МИЭМ включает 1 академика РАН, 4 члена-корреспондента РАН, 34 лауреата государственных премий РФ.

Тесные связи с ведущими отраслевыми институтами, институтами РАН, мировыми компаниями, такими как National Instruments, InfoWatch, Zyxel, QNAP, Altium Limited, а также оснащенные новейшим оборудованием лаборатории: 3D-визуализации; лазерных технологий; телекоммуникации; кибербезопасности — готовить востребованных специалистов на самом высоком уровне.

В МИЭМ НИУ ВШЭ 30 лабораторий — учебных и учебно-исследовательских. Учебные — это лаборатории для занятий по физике, электронике и прочим дисциплинам, а в учебно-исследовательских занимаются научными и проектными разработками. Далее мы покажем вам самые крупные лаборатории.

В МИЭМ НИУ ВШЭ 5 программ бакалавриата и специалитета со стоимостью обучения, варьирующейся от 420 до 540 тысяч рублей. Имеется и 7 программ магистратуры стоимостью обучения в 400 тысяч рублей.

# 3 Описание профессиональных задач

В рамках учебной практики необходимо изучить статью и написать программу по встраиванию водяного знака в изображение. Позже необходимо провести эксперименты, такие как сжатие изображения с водяным знаком различными методами, изменение шага квантования при встраивании, сравнение значений пикселей изображения после встраивания водяного знака при различном шаге квантования и размере с исходным, простое встраивание водяного знака в изображение. На основе этих экспериментов подсчитываются различные метрики качества для изображений.

После выполнения экспериментов и подсчета метрик строится зависимость показателей метрик в таблицах или с помощью графических библиотек в выбранной среде программирования.

Позже делаются выводы по проделанной работе и проведенным экспериментам.

# 4 Исполненное индивидуальное задание

В качестве встраивания информации был предложен алгоритм, основанный на преобразовании Хаара. На вход подается изображение, водяной знак, ключ аффинного преобразования. Водяной знак проходит аффинное преобразование, тем самым повышая безопасность представленной схемы, позже преобразуется в двоичную строку для каждого из каналов RGB. Изображение аналогично разбивается на каналы RGB, позже для каждого канала происходит выбор блока из с-матрицы. С-матрица представляет собой матрицу выбора блоков 2\*2 в изображении, которые не должны пересекаться. Позже на основании преобразования Хаара считается максимальный коэффициент блока, на основании встраиваемого бита в блок 2\*2 изображения и шага квантования подсчитываются верхняя и нижняя граница блока. Либо верхняя граница, либо нижняя присваивается коэффициенту(аналогично на основе шаге квантования). Считается разность между максимальным коэффициентом и получившимся, она же добавляется к каждому пикселю блока. Выбор блока не заканчивается, пока есть бит для встраивания.

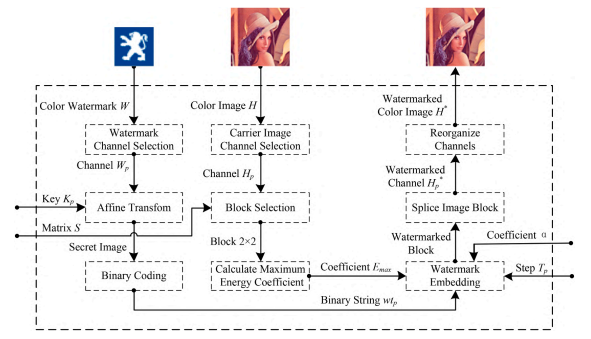


Рисунок 4.1 – алгоритм встраивания водяного знака

Для алгоритма извлечения также выбирается блок 2\*2 из изображения, считается максимальный коэффициент, на основе шага квантования извлекается бит водяного знака.

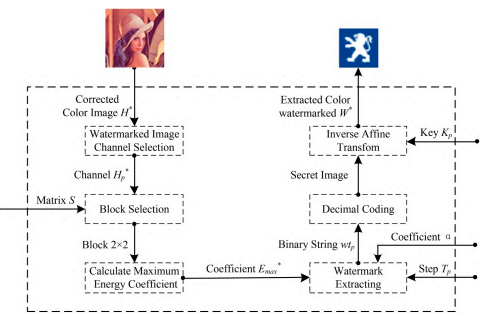


Рисунок 4.2 – алгоритм извлечения водяного знака

# 4.1 Построение программы на языке Python

На вход подается пути к исходному изображению и водяному знаку, задается шаг квантования. Изображение (self.img) представляется в виде трехмерного массива, первые 2 указателя отвечают за расположения пикселя, остальной за хранение значения слоев изображения. Водяной знак (self.wat) представляется в виде одномерного массива с 3 значениями, каждое отвечает за подряд записанную строку двоичных значений всех пикселей для каждого из слоев (red, green, blue). С-матрица (self.s\_mat) представляет собой матрицу блоков 2\*2, посредством которой идет выбор блока для дальнейшего встраивания в него водяного знака.

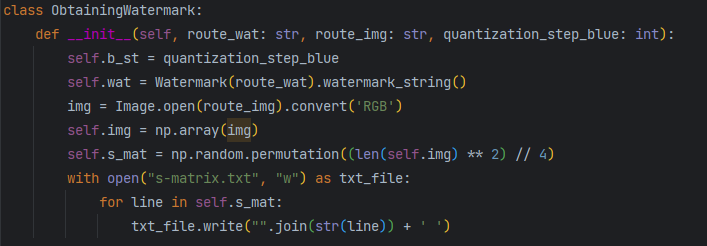


Рисунок 4.1.1 – инициализация класса ObtainingWatermark

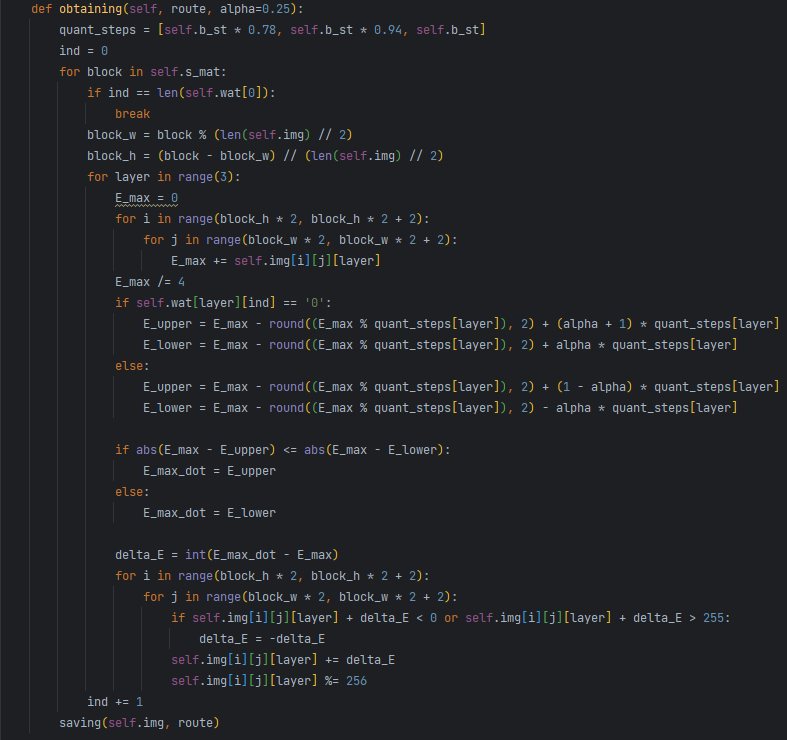
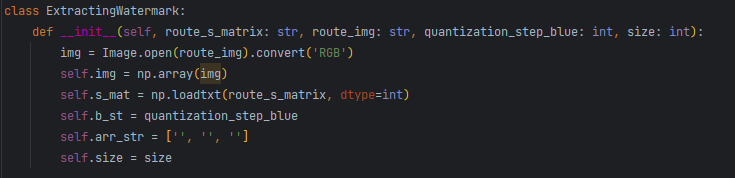
Функция Obtaining, заданная в классе ObtainingWatermark, получает на вход путь для сохранения изображения и коэффициент альфа для встраивания водяного знака. Позже для каждого блока 2\*2 в изображении в зависимости от с-матрицы начинается встраивание информации, пока в self.wat не закончатся необходимые для встраивания биты. Причем кол-во битов, которые можно встроить в изображение <= количеству блоков 2\*2 в изображении. Для каждого блока считается максимальный коэффициент блока (E\_max), в соответствии с преобразованием Хаара он является средним значением пикселей для каждого слоя в данном блоке. Считаются нижняя и верхняя граница максимального коэффициента блока (E\_upper, E\_lower), в зависимости от того, встраиваем ли мы значение бита 1 или 0. Позже ищем оптимальную границу (E\_max\_dot) в зависимости от разности с максимальным значением блока. Для каждого значения пикселя в данном блоке и данном слое прибавляем дельту(delta\_E), равную разности оптимальной границы и максимального коэффициента. В функцию saving передается измененный массив со значениями пикселей (self.img) и путь к сохраненному изображению (route). На этом заканчивается процесс встраивания изображения. 

Рисунок 4.1.2 – встраивание водяного знака в изображение

При инициализации класса извлечения водяного знака из изображения (ExtractingWatermark) на вход подается путь к с-матрице(route\_s\_matrix), путь к изображению(route\_img), шаг квантования(quantization\_step\_blue) и размер водяного знака, находящегося в изображении (size). Изображение аналогично представляется в виде трехмерного массива, задается также массив для извлечения битов водяного знака(self.arr\_str).  Рисунок 4.1.3 – инициализация класса ExtractingWatermark

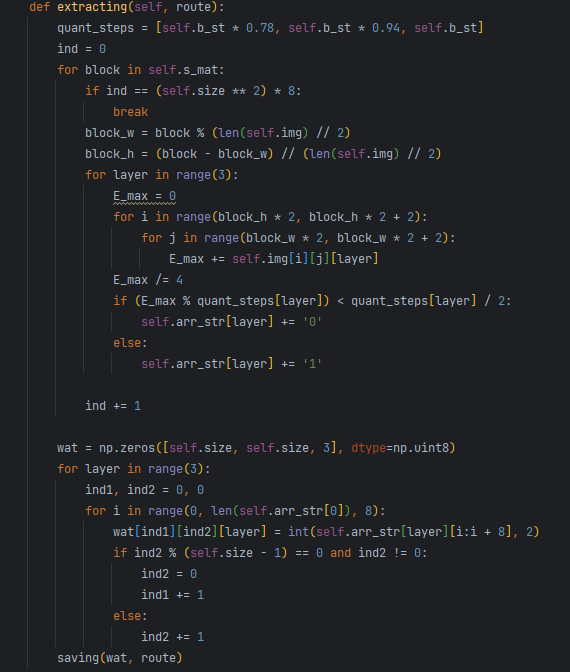
Функция extracting в классе ExtractingWatermark получает на вход путь(route) для сохранения водяного знака. Задаются аналогично функции встраивания шаги квантования (quant\_steps) для каждого из слоев, сделано для того, чтобы человеческий глаз сложнее различал встраивание. Позже для каждого блока изображения мы считаем аналогично максимальный коэффициент (E\_max), его остаток от деления на шаг квантования. Рассматриваем меньше ли он, чем половина от значения шага квантования, если да – то для данного блока и слоя извлеченное значения пикселя – 0, иначе – 1. Добавляем значение в массив водяного знака(self.arr\_str). Позже создаем массив(wat), куда записываются полученные десятичные значения водяного знака. Конвертируем в изображение. 

Рисунок 4.1.4 – встраивание водяного знака в изображение

# 4.2 Эксперименты и подсчет метрик

**4.2.1 Зависимость пикового отношения сигнала к шуму от шага квантования**

В два изображения 512\*512 встраиваются водяные знаки с размерами 90\*90 и 32\*32. Причем каждый водяной знак встраивается с шагом квантования, ранжирующимся от 3 до 40.



Рисунок 4.2.1 – эксперимент с различным шагом квантования

Получаем для двух изображений (4,11) схожие результаты. При увеличении шага квантования отношение сигнала к шуму закономерно убывает. Однако, при встраивании большего числа бит в исходное изображение, отношение сигнала к шуму (PSNR) достигает меньших значений.

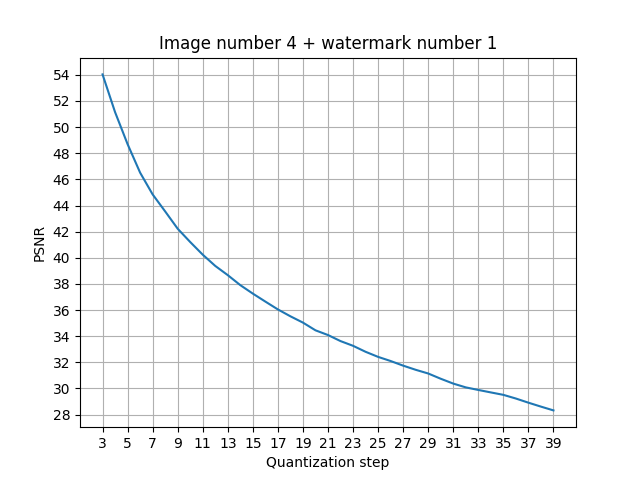


Рисунок 4.2.2 – водяной знак размер 90\*90 и изображение 4

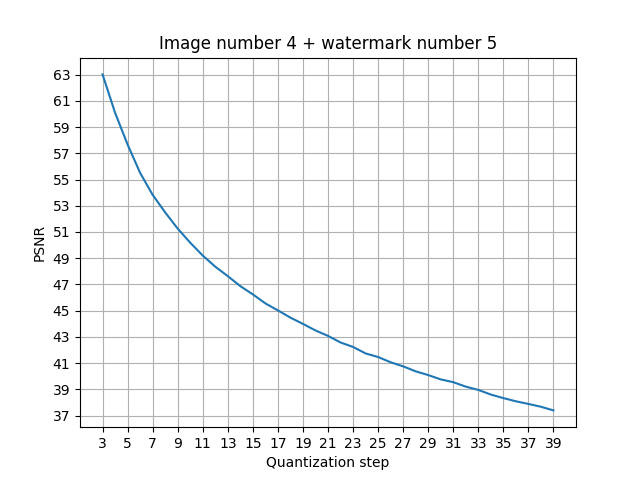


Рисунок 4.2.3 – водяной знак размером 32\*32 и изображение 4

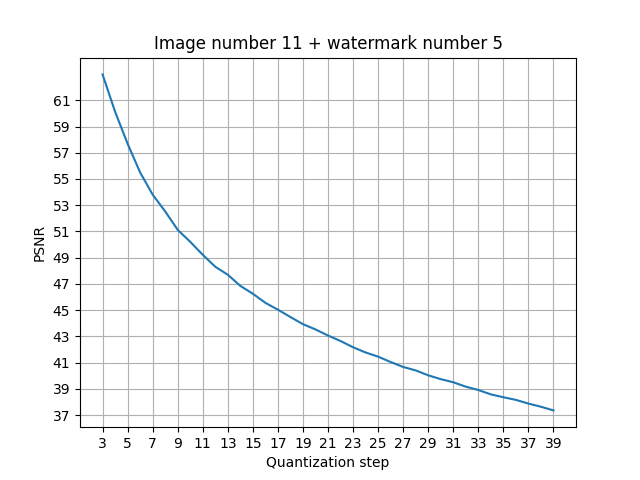
****

Рисунок 4.2.4 – водяной знак размером 32\*32 и изображение 11

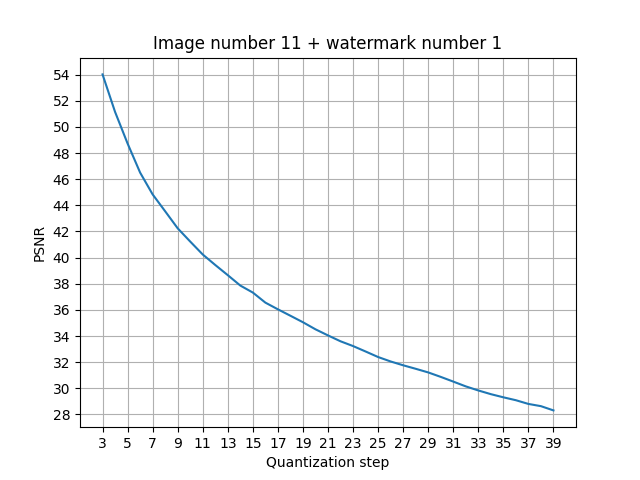
****

Рисунок 4.2.5 – водяной знак размером 90\*90 и изображение 11

**4.2.2 Метрики для различных изображений и водяных знаков**

В одиннадцать изображений 512\*512 встраиваются три водяных знака с размерами 32\*32, шаг квантования равен 24. Подсчитываются метрики PSNR, SSIM, NCC, BER. На основе данных строится таблица.



Рисунок 4.2.6 – метрики для различных изображений.

Таблица 1.1 - метрики для разных изображений и водяных знаков

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изображение | Водяной знак 1 | | | | Водяной знак 2 | | | | Водяной знак 3 | | | |
| SSIM | PSNR | NCC | BER | SSIM | PSNR | NCC | BER | SSIM | PSNR | NCC | BER |
| 1 | 0.997 | 41.858 | 0.999 | 0 | 0.997 | 41.947 | 0.999 | 0 | 0.998 | 41.842 | 0.999 | 0 |
| 2 | 0.999 | 41.740 | 0.999 | 0 | 0.999 | 41.693 | 0.999 | 0.0002 | 0.999 | 41.720 | 0.999 | 0.0001 |
| 3 | 0.999 | 41.769 | 0.999 | 0.0003 | 0.999 | 41.694 | 0.999 | 0.0005 | 0.999 | 41.832 | 0.999 | 0.0005 |
| 4 | 0.999 | 41.804 | 0.999 | 0.0001 | 0.999 | 41.767 | 0.999 | 0 | 0.999 | 41.768 | 0.999 | 0.0002 |
| 5 | 0.999 | 41.791 | 0.999 | 0 | 0.999 | 41.833 | 0.999 | 0 | 0.999 | 41.819 | 0.999 | 0 |
| 6 | 0.999 | 41.821 | 0.999 | 0 | 0.999 | 41.807 | 0.999 | 0 | 0.999 | 41.798 | 0.999 | 0.0001 |
| 7 | 0.997 | 41.752 | 0.999 | 0 | 0.997 | 41.708 | 0.999 | 0 | 0.997 | 41.944 | 0.999 | 0 |
| 8 | 0.997 | 41.828 | 0.999 | 0 | 0.997 | 41.854 | 0.999 | 0 | 0.996 | 41.827 | 0.999 | 0 |
| 9 | 0.997 | 41.728 | 0.999 | 0 | 0.997 | 41.695 | 0.999 | 0 | 0.998 | 41.786 | 0.999 | 0 |
| 10 | 0.999 | 41.812 | 0.999 | 0 | 0.999 | 41.840 | 0.999 | 0 | 0.999 | 41.780 | 0.999 | 0 |
| 11 | 0.999 | 41.764 | 0.999 | 0.0005 | 0.999 | 41.765 | 0.999 | 0.0005 | 0.999 | 41.813 | 0.999 | 0.0008 |

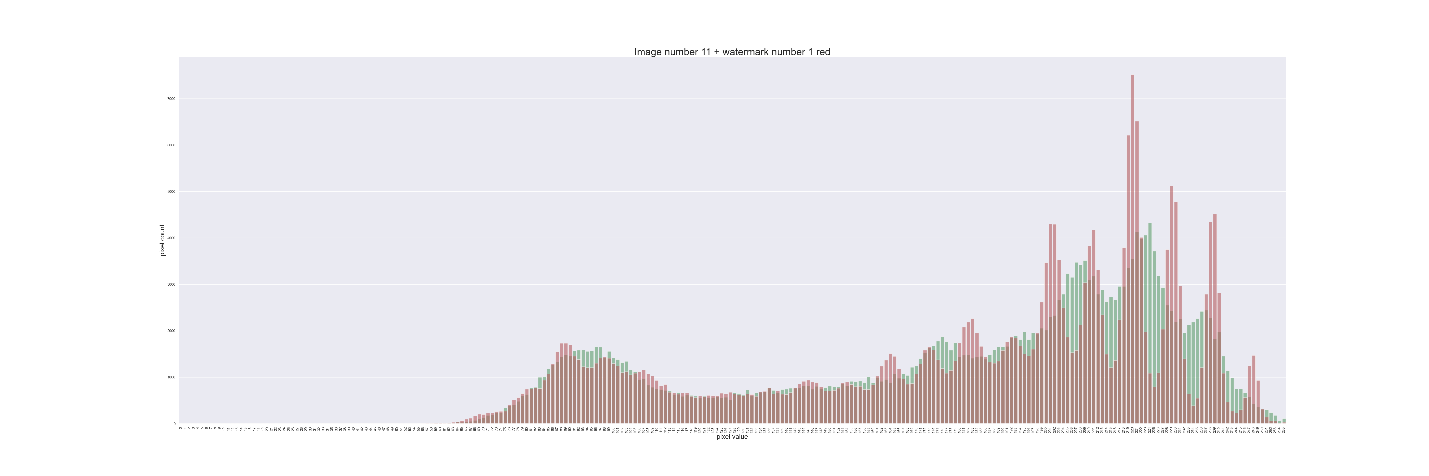
Для каждого изображения и водяного знака получаем примерно равные значения, можно сделать вывод, что алгоритм работает для всех изображений и водяных знаков с приблизительно равной точностью.

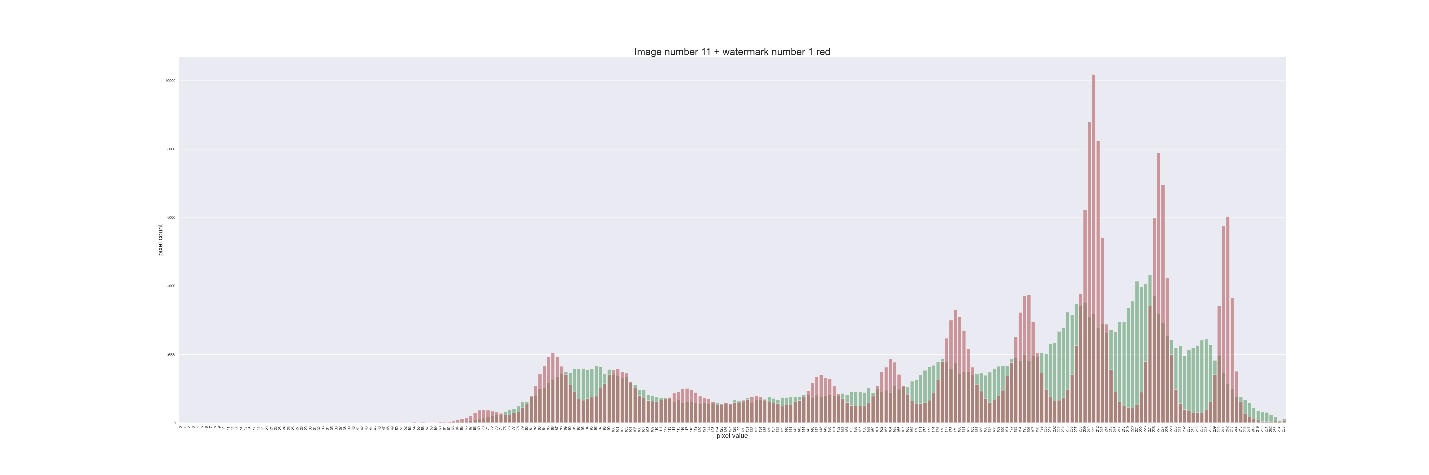
**4.2.3 Зависимость значений пикселей после встраивания водяного знака**

В изображение встраивается разные водяные знаки с различным шагом квантования. Нужно построить зависимость значений пикселей до и после встраивания.

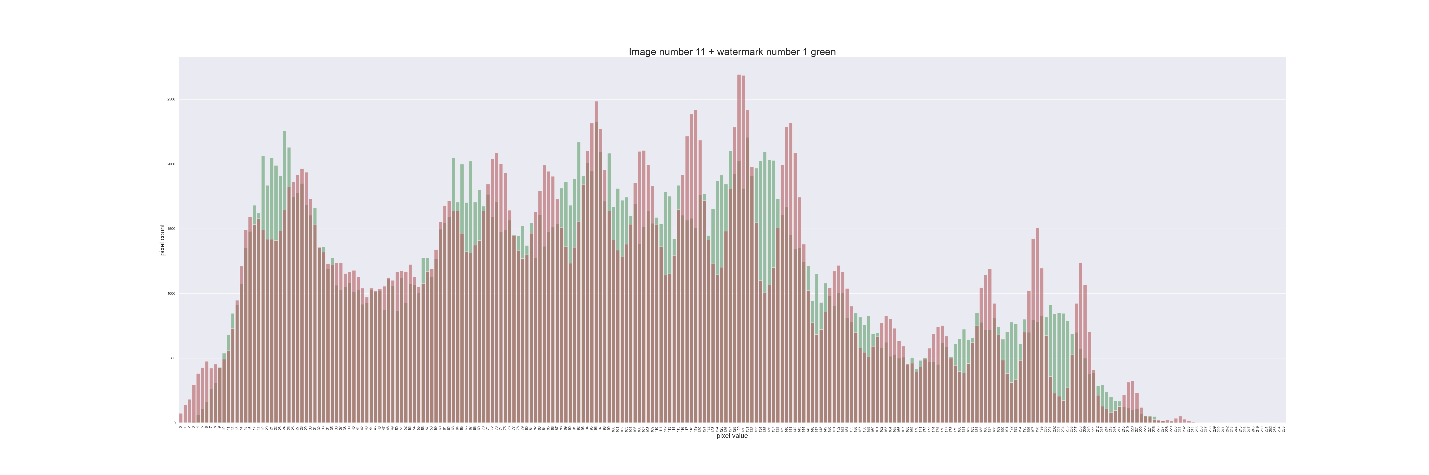
 Рисунок 4.2.7 – значения пикселей до/после встраивания

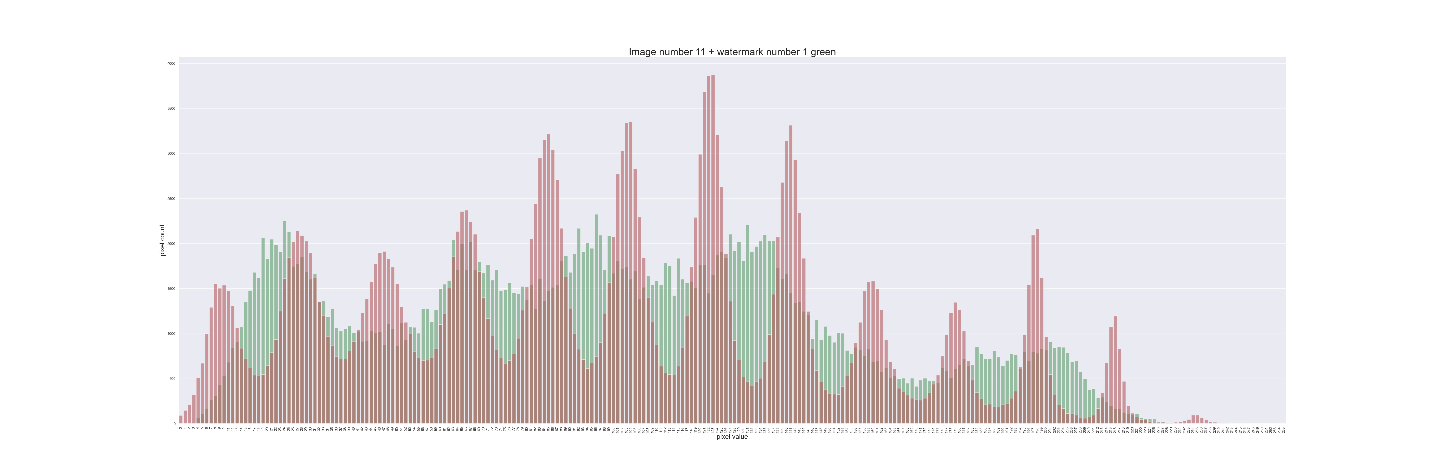
Получаем для встраивания большого водяного знака с шагом квантования 24/40.

Красный слой:Рисунок 4.2.8 – шаг квантования 24, красный слой, большой водяной знак

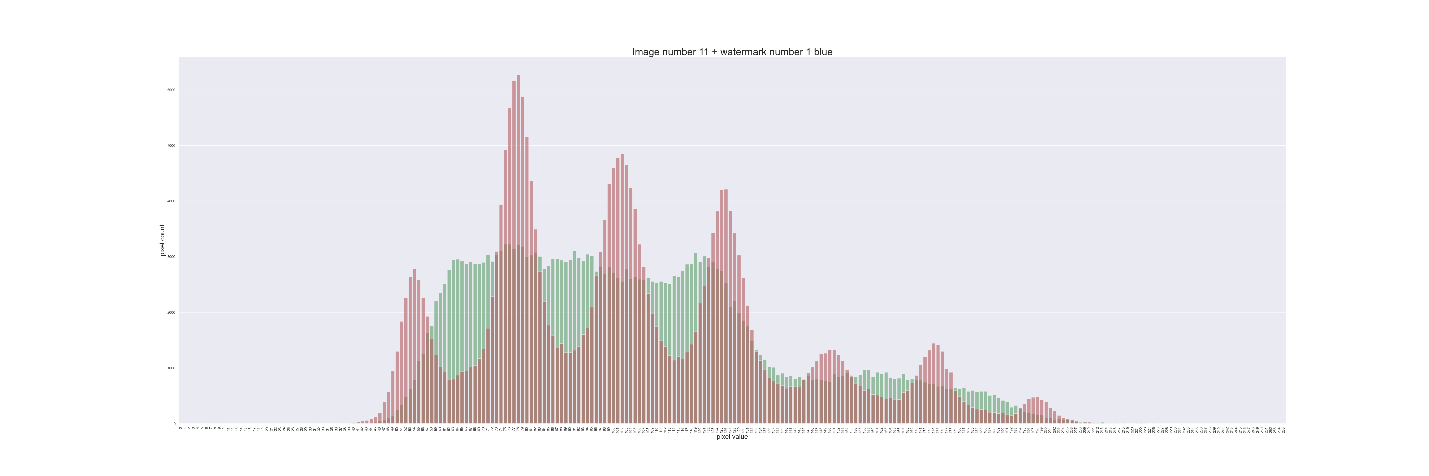
 Рисунок 4.2.9 – шаг квантования 40, красный слой, большой водяной знак

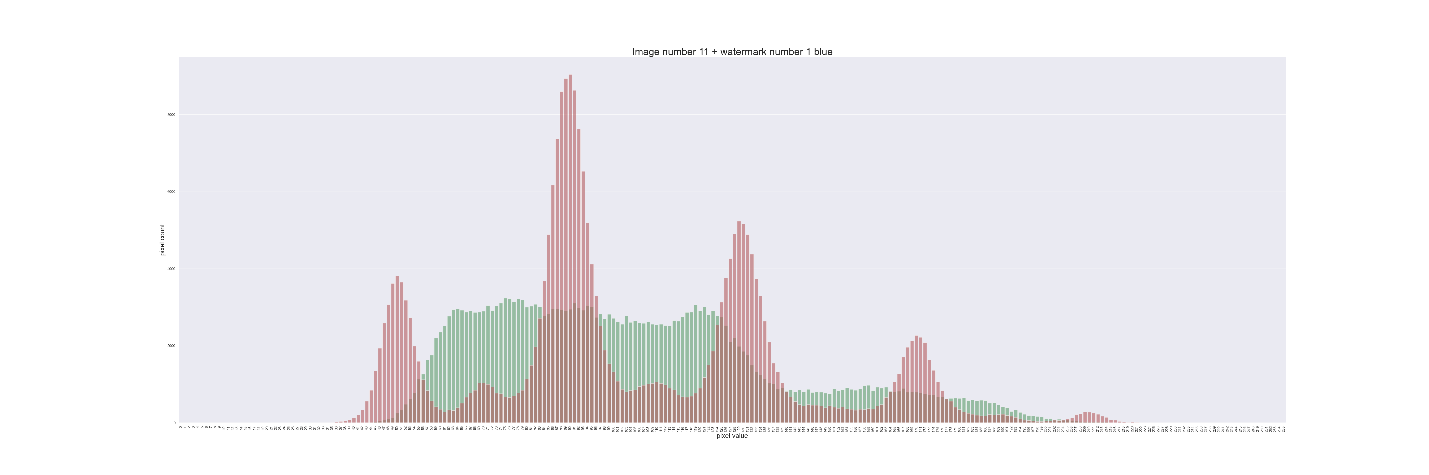
Зеленый слой:

****Рисунок 4.2.10 – шаг квантования 24, зеленый слой, большой водяной знак

**** Рисунок 4.2.11 – шаг квантования 40, зеленый слой, большой водяной знак

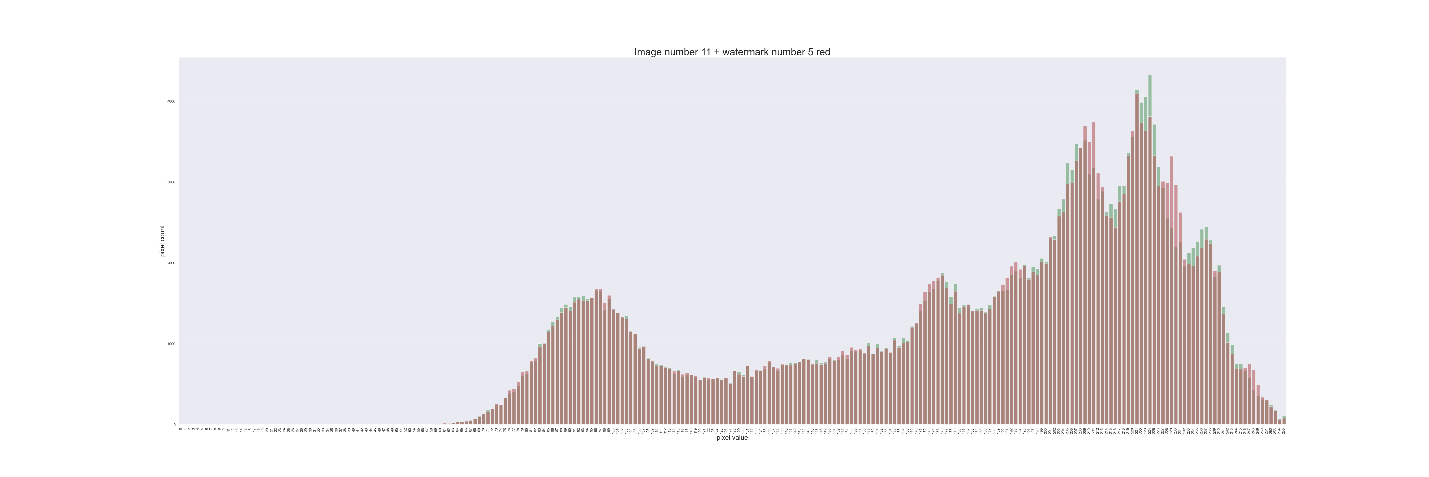
Синий слой:

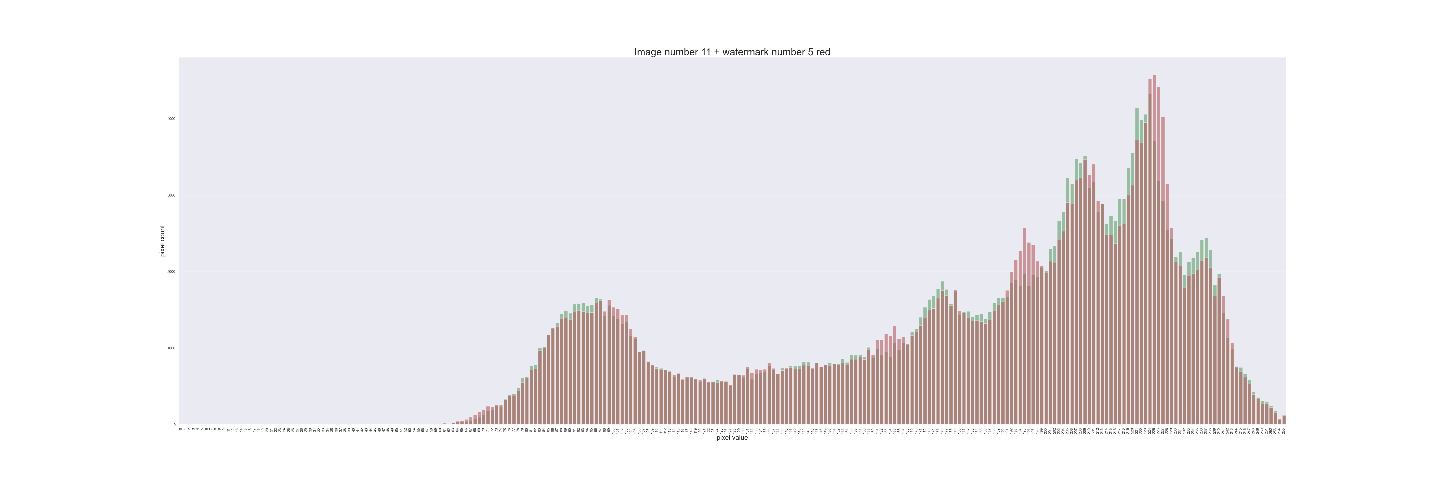
**** Рисунок 4.2.12 – шаг квантования 24, синий слой, большой водяной знак

****Рисунок 4.2.13 – шаг квантования 40, синий слой, большой водяной знак

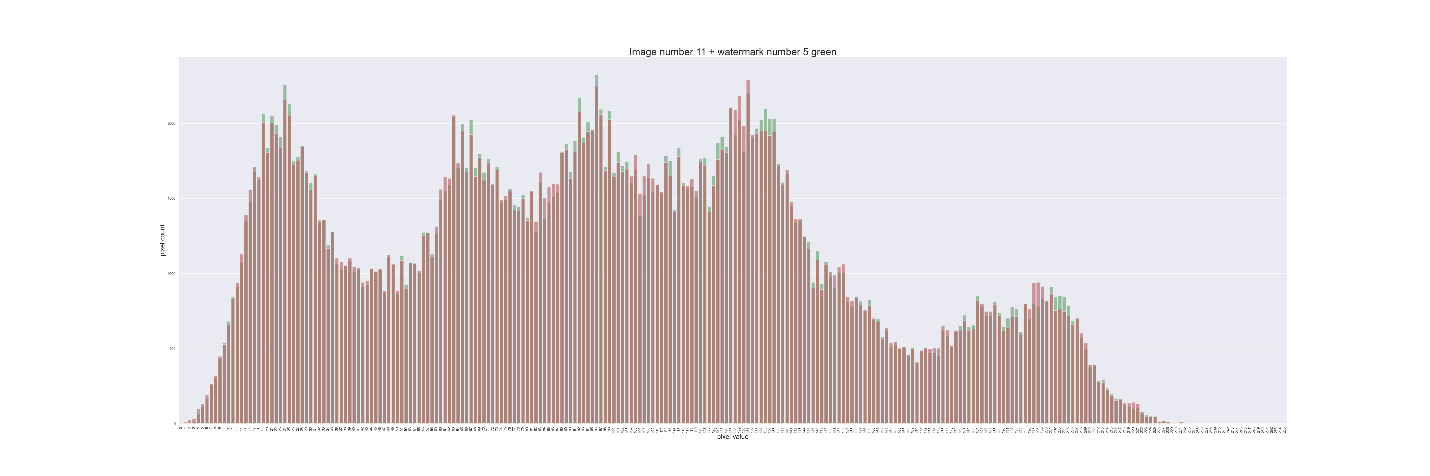
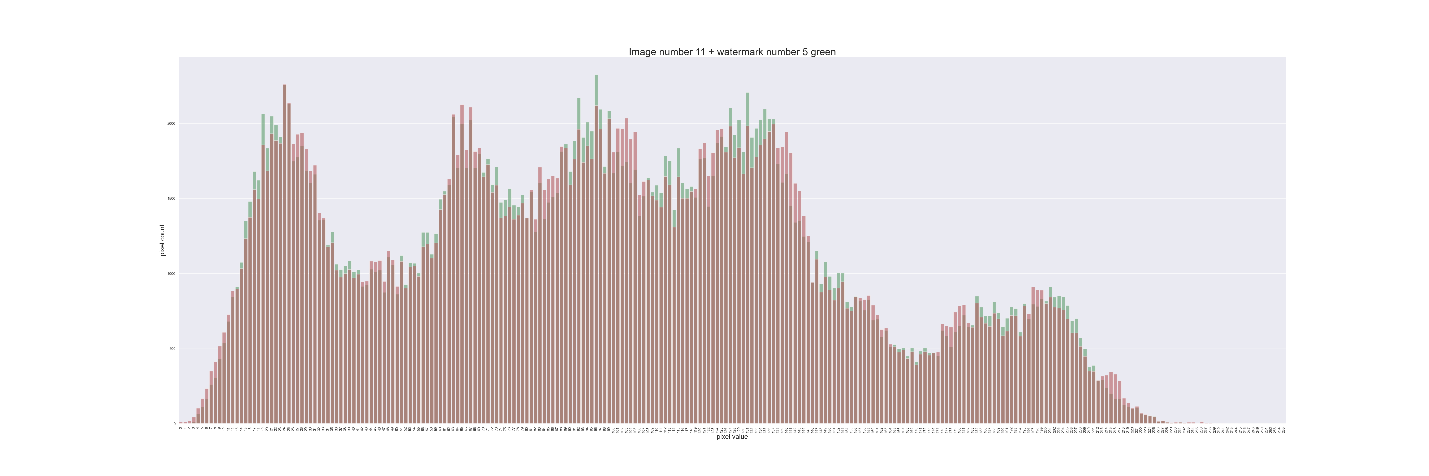
Получаем для встраивания малого водяного знака с шагом квантования 24/40.

Красный слой:

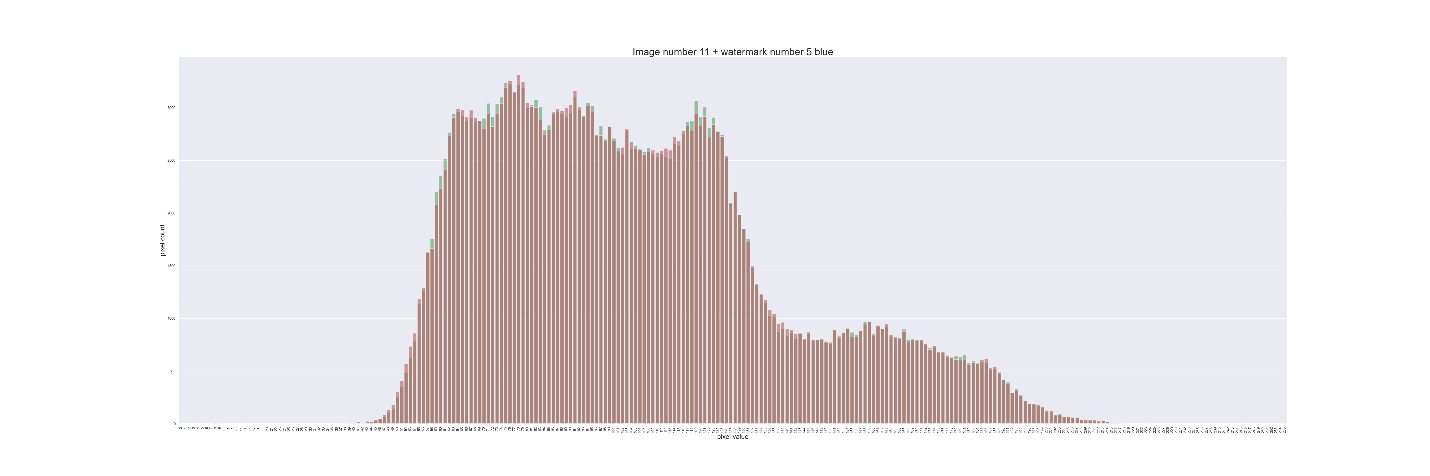
Рисунок 4.2.14 – шаг квантования 24, красный слой, малый водяной знак

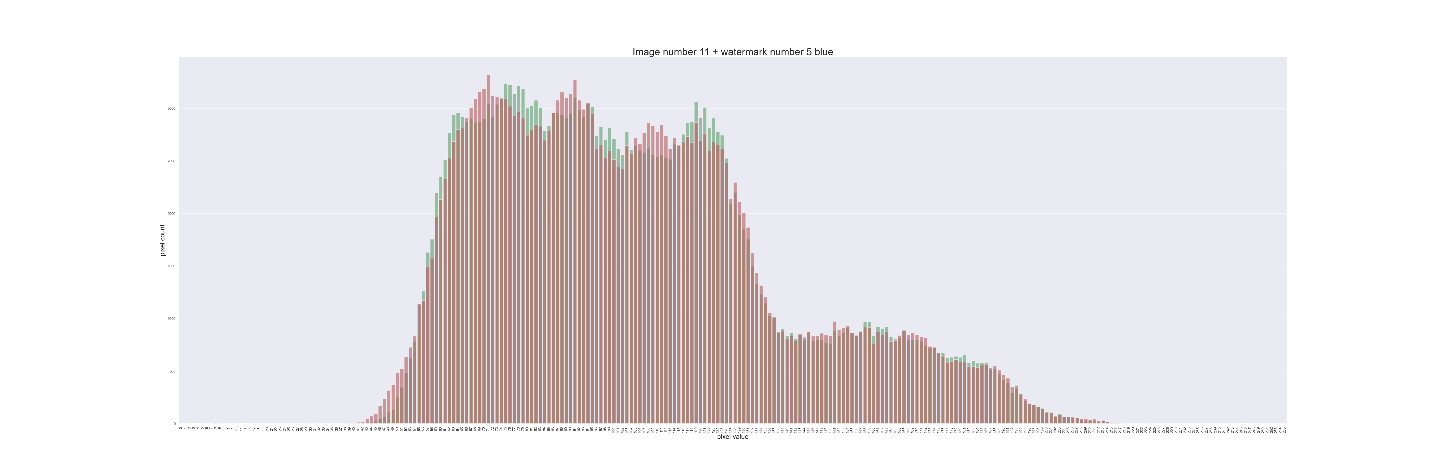
 Рисунок 4.2.15 – шаг квантования 40, красный слой, малый водяной знак

Зеленый слой:

 Рисунок 4.2.16 – шаг квантования 24, зеленый слой, малый водяной знак Рисунок 4.2.17 – шаг квантования 40, зеленый слой, малый водяной знак

Синий слой:

 Рисунок 4.2.18 – шаг квантования 24, синий слой, малый водяной знак

**** Рисунок 4.2.19 – шаг квантования 40, синий слой, малый водяной знак

Получаем такие графики зависимости значения пикселя от числа пикселей, где зеленый фон – количество пикселей до встраивания, красный фон – количество пикселей после встраивания.

После данного эксперимента можно сделать вывод о том, что при увеличении шага квантования число одинаковых пикселей возрастает, причем проявляется этой в большей мере на красном слое, позже на зеленом, а после на синем. Также можно заметить, что при увеличении размера встраиваемого изображения меняется количество на каждом значении пикселя в большую сторону, распределение по различным пикселям становится более заметно.

**4.2.4 Сжатие изображения с водяным знаком методом JPEG**

В эксперименте необходимо сжать одно изображение с водяным знаком методом JPEG и посчитать метрики NCC и BER для водяного знака. Изображения сжимается с параметрами качества равным 100,80,60,40,20.

 Рисунок 4.2.20 – сжатие изображения jpeg

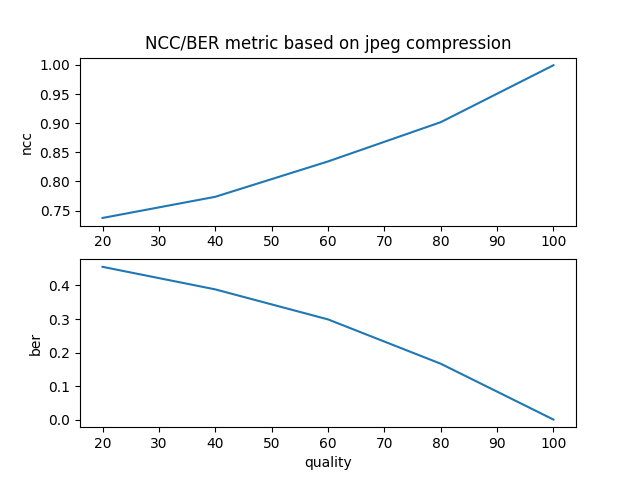


Рисунок 4.2.22 – результаты сжатия изображения

В ходе эксперимента установлено, что по увеличению качества сжатия изображения уменьшается ошибка битов пикселей водяного знака при извлечении и увеличивается метрика NCC.

**4.2.5 Сжатие изображения с водяным знаком методом JPEG2000**

В эксперименте необходимо сжать изображения с водяным знаком с помощью метода JPEG2000, позже извлечь водяной знак и посчитать NCC/BER метрики для него.

 Рисунок 4.2.23 – сжатие изображение JPEG2000

Таблица 1.2 - метрики NCC/BER разных изображений, водяных знаков после JPEG2000

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Изображение | Водяной знак 1 | | Водяной знак 2 | | |
| NCC | BER | NCC | BER |
| 1 | 1.0 | 0 | 1.0 | 0 |
| 2 | 0.99998 | 0.00009 | 0.99994 | 0.00040 |
| 3 | 0.99994 | 0.00024 | 0.99976 | 0.00052 |
| 4 | 0.99995 | 0.00021 | 0.99998 | 0.00016 |
| 5 | 1.0 | 0 | 1.0 | 0 |
| 6 | 0.99998 | 0.00008 | 0.99995 | 0.00012 |
| 7 | 1.0 | 0 | 1.0 | 0 |
| 8 | 1.0 | 0 | 1.0 | 0 |
| 9 | 1.0 | 0 | 1.0 | 0 |
| 10 | 1.0 | 0 | 1.0 | 0 |
| 11 | 0.99977 | 0.00088 | 0.99986 | 0.00081 |

В эксперименте получаются практически идеальные метрики, представленный метод может противодействовать JPEG2000 сжатию.

# 5 Заключение

В ходе практики была написана программа по встраиванию/извлечению водяного знака из изображения. Программа и все эксперименты были загружены на мой GitHub (<https://github.com/metobtw/image_watermarking>). Было проведено несколько экспериментов, в ходе которых установлено, что программа хорошо извлекает водяной знак при JPEG2000 сжатии, при JPEG, если значение качества ранжируется от 60 и выше. Также были измерена метрика PSNR при встраивании разного кол-ва бит в изображение, от полного заполнения, до частичного. Рассмотрено распределение по значениям пикселей до/после встраивания.

# Список использованных источников

1. SIPI Image Database – URL: <https://sipi.usc.edu/database/database.php> (дата обращения 27.02.2023)

2. SSIM – Википедия – URL: <https://ru.wikipedia.org/wiki/SSIM> (дата обращения 27.02.2023)

3. Decheng Liu, Qingtang Su, Zihan Yuan, Xueting Zhang A fusion-domain color image watermarking based on Haar transform and image correction// Expert Systems with Applications – 2021 –N 170

4. Пиковое отношение сигнала к шуму – Википедия – URL: <https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D0%BA%D0%BE%D0%B2%D0%BE%D0%B5_%D0%BE%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB%D0%B0_%D0%BA_%D1%88%D1%83%D0%BC%D1%83> (дата обращения 27.02.2023)

5. Pillow (PIL Fork) 9.4.0 documentation – ReadTheDocs – URL: <https://pillow.readthedocs.io/en/stable/> (дата обращения 20.02.2023)

6. Московский институт электроники и математики им. А.Н. Тихонова – URL: <https://miem.hse.ru/> (дата обращения 22.03.2023)