МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное образовательное учреждение

высшего образования

**«Национальный исследовательский**

**Нижегородский государственный университет им. Н.И. Лобачевского»**

**Институт информационных технологий, математики и механики**

Направление подготовки «Фундаментальная информатика и информационные технологии»

**ОТЧЕТ**

по лабораторной работе

**Устранение шума на фотографиях, полученных при съёмке на микроскоп**

**Выполнил:** студент группы

381906-2 Углиснкий Б.С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Подпись

**Проверил:** Гетьманская А.А

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород

2021

Оглавление

[Введение 3](#_Toc89368442)

[1.Постановка задачи 4](#_Toc89368443)

[2.Руководство пользователя 5](#_Toc89368444)

[3. Руководство программиста 6](#_Toc89368445)

[3.1 Описание структуры программы 6](#_Toc89368446)

[3.2 Описание работы программы 6](#_Toc89368447)

[3.3 Описание программной реализации 6](#_Toc89368448)

[4. Результаты экспериментов 8](#_Toc89368449)

[5. Заключение 12](#_Toc89368450)

[Список Литературы 13](#_Toc89368451)

Введение

Данная лабораторная работа направлена на изучение способа устранения шума на изображении при использовании преобразований Фурье. Программа должна уметь переводить изображения в частотный спектр, при помощи быстрого преобразования Фурье, а также выявлять периодический шум и удалять его. Основная цель данной лабораторной работы выяснить, применим ли этот способ на данном датасете. Необходимо провести исследование с различными параметрами определения высокочастотных участков и выяснить какой из них самый эффективный.

1.Постановка задачи

*Цель:*

Разработать и реализовать программу, выявляющую периодический шум на изображении, с последующим его удалением.

Шум на изображении - нежелательный побочный продукт захвата изображения, который скрывает желаемую информацию. Обычным источником периодического шума в изображении являются электрические или электромеханические помехи, во время процесса захвата изображения. Также причиной периодического шума может быть неисправное оборудование или ошибки людей, использующие его. Изображение, подверженное периодическому шуму, будет выглядеть так, как будто поверх исходного изображения был добавлен повторяющийся узор. В частотной области этот тип шума можно увидеть в виде дискретных всплесков. Существенного снижения этого шума можно добиться, применяя режекторные фильтры в частотной области.

Для написания программы используется язык Python

*Исходные данные:*

Набор из 13 зашумлённых изображений размером 2048x1362 пикселей

*Требуемый результат:*

Изображения с отфильтрованным шумом

2.Руководство пользователя

1. Для начала работы необходимо открыть файлы Research.ipynb.
2. Запустить на исполнение все ячейки.
3. Во время работы программы пользователю будет показываться результат обработки каждого изображения, а также его частотный спектр.
4. Результаты работы можно будет посмотреть в соответствующих папках:
   1. DFFT\_Compare - сравнения исходного изображения с его частотным спектром
   2. DFFT\_Images - исключительно частотный спектр картинок
   3. Notch - изображения после зануления самых высокочастотных участков
   4. Gauss\_Q - для изображений сначала был применён фильтр Гаусса, а затем Notch фильтр.

**3. Руководство программиста**

3.1 Описание структуры программы

Программа состоит из 4 файлов с расширением .py: DFFT.py, Filters.py, Filters.py, Visualisation.py и одного питоновского ноутбука: Research.ipynb

3.2 Описание работы программы

При запуске Research.ipynb программа по очереди применяет различные фильтры на изображения. Сначала работает частотный фильтр Notch, а затем комбинация фильтра Гауcса и Notch.

***Алгоритмы***

Цикл обработки изображения состоит из следующих этапов:

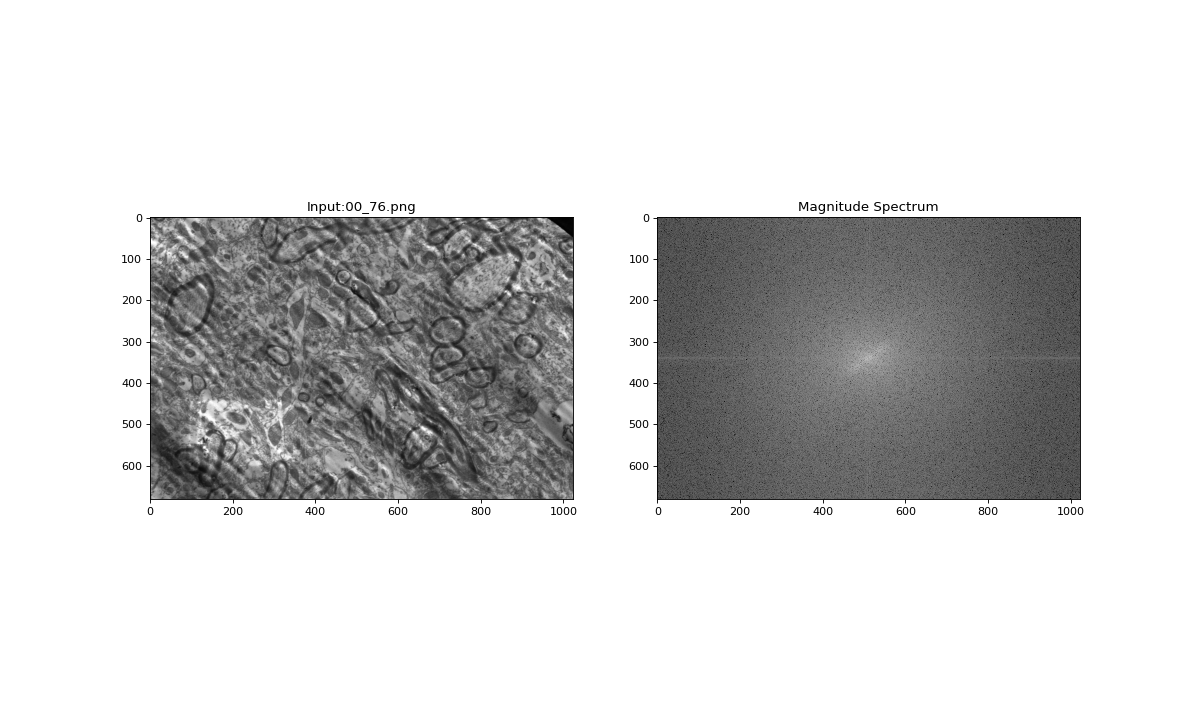
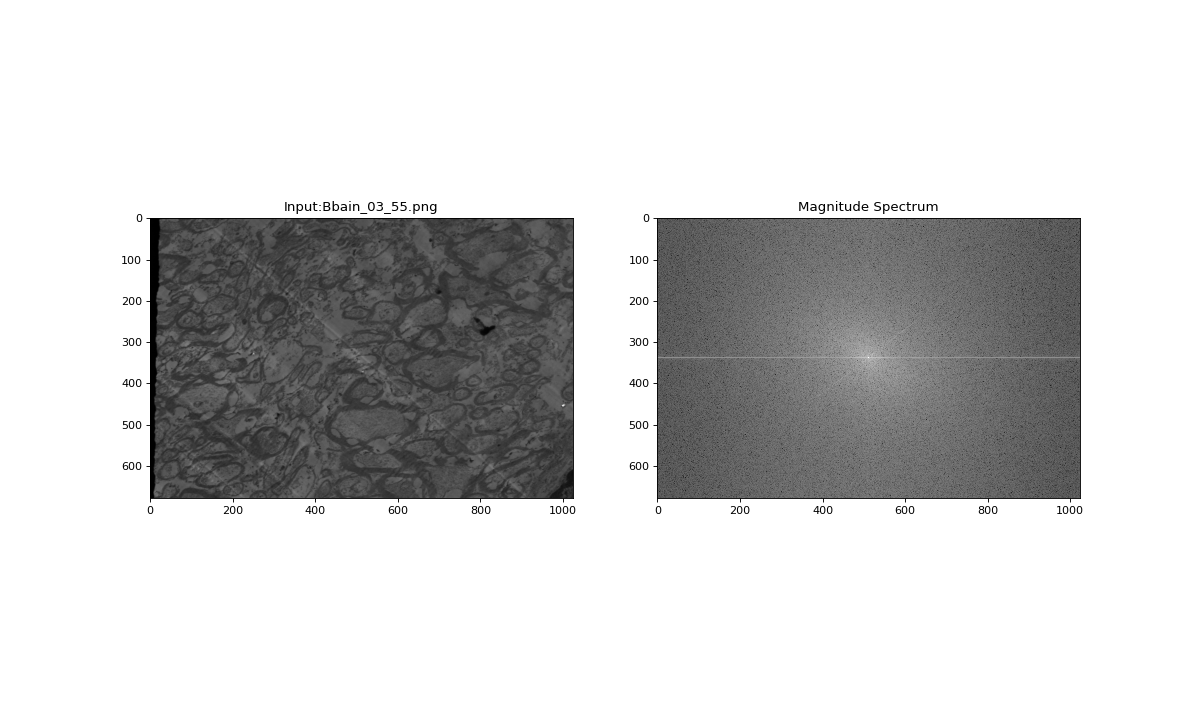
1. Применение быстрых преобразований Фурье с последующим сдвигом высокочастотных компонент в центр матрицы частот.
2. Нахождение магнитуды каждого значения матрицы
3. Применение низкочастотного фильтра.
4. Использование обратного преобразования Фурье для получения результирующего изображения.

3.3 Описание программной реализации

* **DFFT.py** - Содержит функции по применению преобразований Фурье:
  + ApplyDFFT - применение быстрых преобразований Фурье.
  + GistDFFT - преобразовывает результат преобразований Фурье, для представления в виде картинки, достаточно информативной, для применения при последующих исследованиях.
  + ReverseDFFT - осуществляет обратное преобразование Фурье.
* **FileWorker.py** - содержит класс, который осуществляет чтение и запись изображений:
  + ReadAll - возвращает названия всех изображений в директории.
  + ReadImg - возвращает картинку в виде numpy массива.
  + SaveImg –сохраняет изображения на устройство долговременного хранения.
* **Filters.py** - реализует фильтры для обработки изображений:
  + Gauss - фильтр Гаусса.
  + Notch - частотный фильтр.
* **Visualisation.py** - предоставляет интерфейс, для визуализации и демонстрации полученных в результате работы программы изображений.
  + ShowDFFT - позволяет увидеть результат DFFT.
  + CompareImages - позволяет сравнить любых два изображения, с последующим сохранением результата.

4. Результаты экспериментов

*DFFT:*

Первым делом изображения были преобразованы в частотный спектр. Далее было необходимо применить логарифмирование для каждого значения, чтобы нивелировать большую разность в значениях частот и детальнее рассмотреть полученные образцы: 

Как мы можем видеть, в полученных частотных спектрах отсутствуют значительные пики, исключая центр. Т.к. периодический шум проявляется в виде вспышек, а в нашем случае они отсутствуют, то можем сделать, что применение частотного фильтра, как мы дальше убедимся, малоэффективно и малоприменимо на данном датасете. Весь шум как бы является частью изображения и, если попробовать снизить планку, при которой мы зануляем частоту, то в результате получим сильно искажённое изображение. Если же повысить её, то изображение толком и не измениться, т.к. отбрасывать будет попросту нечего, кроме центра картинки, но он отвечает за саму рельефность картинки, и его убирать нельзя.

*Notch:*

Низкочастотный фильтра работает следующим образом:

Notch(img,q,dmz=25,hazard\_zone=5):

img-исходное изображение;

q - значение квантили;

dmz (demilitarized zone) - радиус окрестности центра (в доли от изображения);

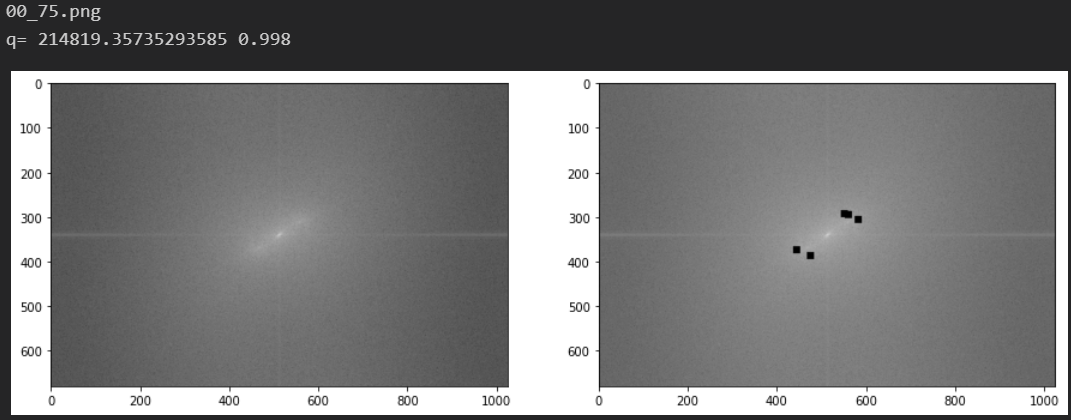
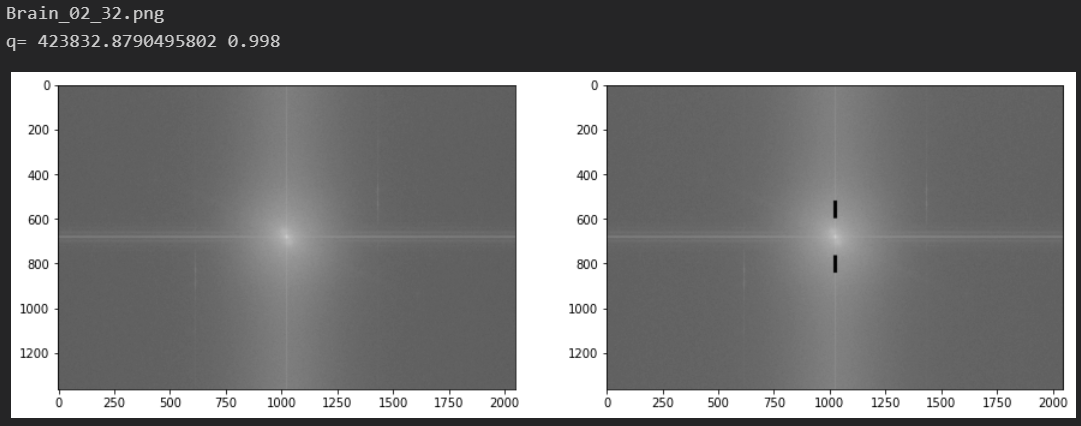
hazard\_zone - радиус зануляемой окрестности (в пикселях);

1. Применяется DFFT
2. Высчитывается квантиль, начиная с которой следует занулять частоты. Так как величины растут довольно быстро, то необходима достаточно большая квантиль (q=0.998), чтобы не испортить изображение.
3. Далее находятся все точки, в которых магнитуда выше, чем квантиль, с последующим занулением значений в небольшой окрестности точки

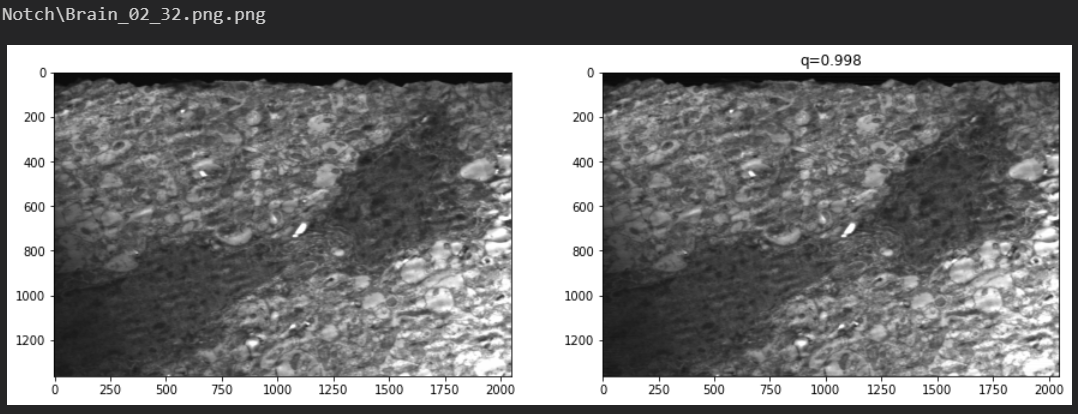
Точки, которые находятся в непосредственной близости к центру частотного спектра, не зануляются, дабы не исказить результат.

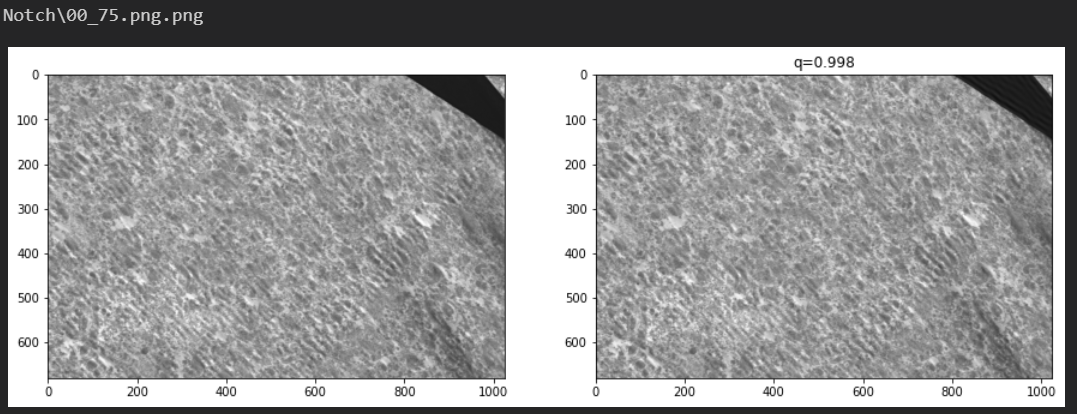
Найти баланс в параметрах достаточно трудно, т.к., либо картинка сильно искажается, либо не находятся высокочастотные участки.

*Пример работы программы:*



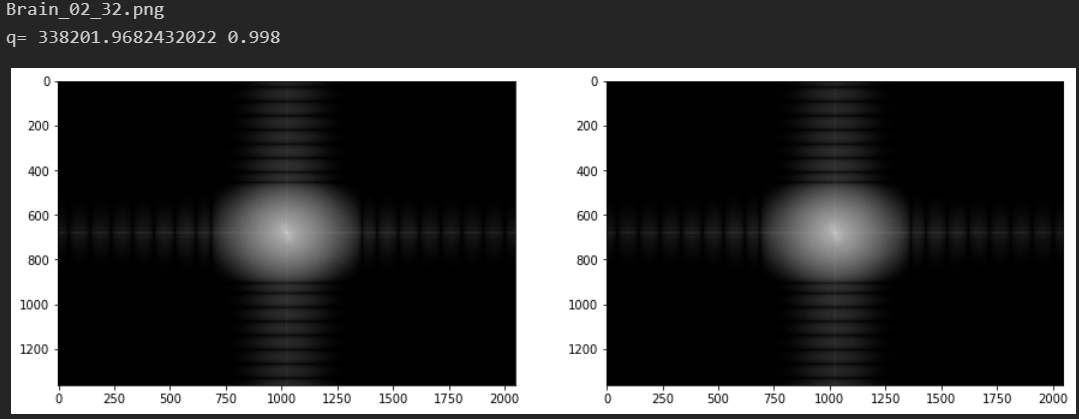
К сожалению, из-за вышеупомянутых причин, каких-то значительных результатов это не принесло:

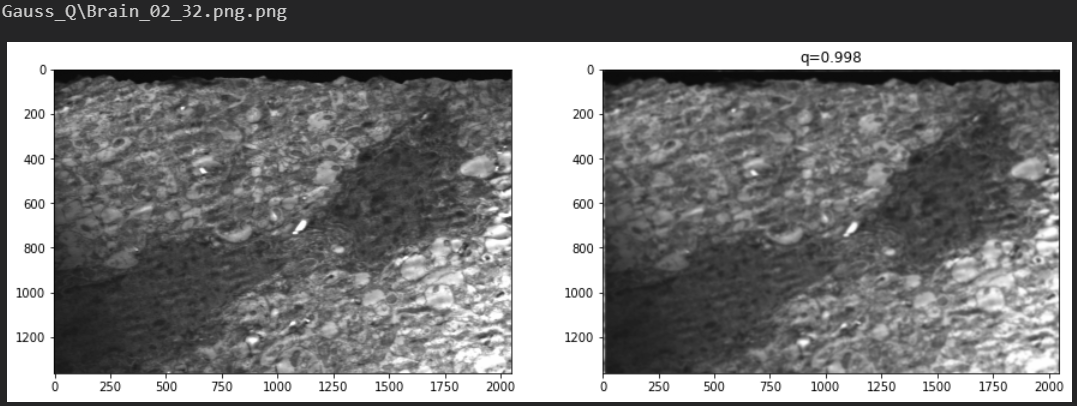




*Gauss + Notch:*

Из-за того, что, как выяснилось, фильтр Гаусса существенно уменьшает значения частот на периферии частотного спектра, последующее применение Notch фильтра оказалось бесполезным:





5. Заключение

В лабораторной работе была реализована программа, по применению низкочастотного фильтра Notch, с возможностью установки параметров: квантили, окрестности зануления значений и окрестности центра. Был применён как фильтр Notch в одиночку, так и его комбинация с фильтром Гаусса. В результате экспериментов, было выяснено, что данный датасет плохо поддаётся частотным фильтрам, из-за своей однородности в частотном спектре и отсутствия дискретных пиков частот. Применение фильтров, ожидаемо, не дало никаких результатов.

Список Литературы

1. Гетманская Александра Александровна - Презентация к учебному курсу «Обработка изображений» “Преобразование Фурье в обработке изображений”
2. Youtube канал 3Blue1Brown – Что такое преобразование Фурье? Иллюстрированное введение. – [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/watch?v=spUNpyF58BY&t=837s>
3. Data Science Central – Solving Some Image Processing Problems with Python libraries - Part 2 – [Электронный ресурс]. – Режим доступа: <https://www.datasciencecentral.com/profiles/blogs/solving-some-image-processing-problems-with-python-libraries-part>