# Предыстория...

Давным-давно, в далекой стране, жили два друга — юный ученый Арнольд и его верный спутник, талантливый художник Леонардо. Они путешествовали по миру, изучая тайны природы и искусства. Однажды, блуждая по густым лесам, они натолкнулись на древнюю башню, окутанную таинственным туманом.

Башня выглядела заброшенной, но внутри слышались странные звуки. Друзья решили войти, ведь любопытство всегда было их слабостью. Как только они переступили порог, дверь захлопнулась за ними, и они оказались заперты внутри.

— Кажется, мы попали в ловушку! — воскликнул Арнольд. — Но почему? Что здесь такого ценного? — спросил Леонардо, оглядываясь вокруг.

Вдруг перед ними появился силуэт старика в длинном плаще. Его голос звучал, словно эхо:

— Добро пожаловать в мою башню, смельчаки! Я — великий магистр изображений, и вы стали моими пленниками. Чтобы выбраться отсюда, вам предстоит пройти испытание.

Старик махнул рукой, и перед друзьями появились две стопки картинок. Одни были четкими и яркими, другие — размытыми и тусклыми.

— Ваша задача проста: определить, кто изображен на каждой картинке, используя свои знания и умения. Но будьте осторожны — некоторые изображения обманчивы!

Арнольд и Леонардо переглянулись. Они понимали, что это не просто игра. Их свобода зависела от успеха в этой задаче.

Так началось их путешествие по миру изображений, полное загадок и неожиданных открытий.

# Введение

Данная работа предполагает активное изучение дополнительных материалов по теме, обсуждения в чате и не только :) В условии собраны основные идеи и требования того, что нужно выполнить. Если возникают вопросы - пишите в чат. Возможно, кто-то уже смог решить вашу проблему и сможет помочь.

Успехов!

## Основная часть

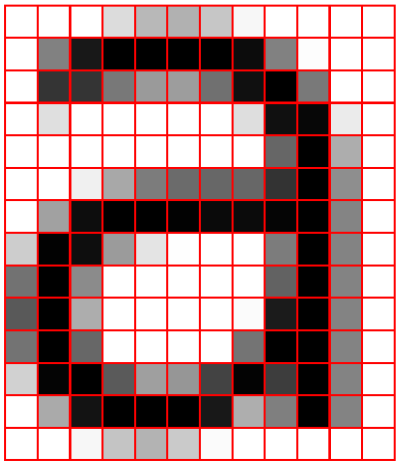
1. Выбрать цветное изображение (формат .jpg, .jpeg, .png) для работы и скачать его.

   Обратите внимание, что вам придется обрабатывать изображение, которое определяется матрицей цветов. Чем больше размер изображения, тем дольше будет обработка.

2. Цветное изображение можно представить как функцию $f : \mathbb{R}^2 \to \mathbb{R}^3$. Если изображение черно-белое (далее будем называть такие серыми), то это функция $f : \mathbb{R}^2 \to \mathbb{R}$.

   Можно подходить с формату хранения яркости пикселей по-разному. Чаще всего, это целое число от 0 до 255. Также можно нормировать все каналы и хранить числа в границах отрезка $[0, 1]$.

# Растровое изображение

a 

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.0 | 1.0 | 1.0 | 0.9 | 0.6 | 0.6 | 0.6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 1.0 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | 1.0 | 1.0 | 1.0 |
| 1.0 | 0.2 | 0.2 | 0.5 | 0.6 | 0.6 | 0.5 | 0.0 | 0.0 | 0.5 | 1.0 | 1.0 |
| 1.0 | 0.9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.9 | 0.0 | 0.0 | 0.9 | 1.0 |
| 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.5 | 0.0 | 0.5 | 1.0 |
| 1.0 | 1.0 | 1.0 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.4 | 0.0 | 0.5 | 1.0 |
| 1.0 | 0.4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.5 | 1.0 |
| 0.9 | 0.0 | 0.0 | 0.6 | 1.0 | 1.0 | 1.0 | 1.0 | 0.5 | 0.0 | 0.5 | 1.0 |
| 0.5 | 0.0 | 0.6 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.5 | 0.0 | 0.5 | 1.0 |
| 0.5 | 0.0 | 0.7 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.5 | 1.0 |
| 0.6 | 0.0 | 0.6 | 1.0 | 1.0 | 1.0 | 1.0 | 0.5 | 0.0 | 0.0 | 0.5 | 1.0 |
| 0.9 | 0.1 | 0.0 | 0.6 | 0.7 | 0.7 | 0.5 | 0.0 | 0.5 | 0.0 | 0.5 | 1.0 |
| 1.0 | 0.7 | 0.1 | 0.0 | 0.0 | 0.0 | 0.1 | 0.9 | 0.8 | 0.0 | 0.5 | 1.0 |
| 1.0 | 1.0 | 1.0 | 0.8 | 0.8 | 0.9 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}.$$

$$0 \le f(x,y) \le L, \quad \text{and typically} \quad L = 255$$

Осуществите разделение изображения по цветовым каналам. Таким образом, у вас будет несколько двумерных массивов, который отвечает за цветовую палитру.

3. Изобразите все пространственные области изображения по отдельности (R, G, B) и парные комбинации (RG, GB, RB).

4. Создайте на основе данного изображение серое изображение.

5. Выберите несколько строк матрицы изображения и постройте графики изменения частоты каналов (для цветного и серого изображений). Частоту для цветных изображений необходимо построить в одной системе координат.

6. Определим свертку функции так

$$g(x,y) = w * f(x,y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} w(i,j) \cdot f(x-i, y-j)$$

где $g$ - изображение после свертки, $w$ - ядро свертки, $f$ - исходное изображение.

Для обработки изображения на краях необходимо определить изображения за пределами "видимых"границ Для этого, можно использовать разные методы. Например, продолжить границы картинок на недостающий размер охвата свертки. Рекомендуем после экспериментов выбрать тот, который хуже всего искажает ваше изображение при обработке.

7. Используя известные ядра преобразований, реализовать алгоритм свертки изображения с примирением фильтра

   (a) Пороговая фильтрация
   (b) Медианный фильтр
   (c) Фильтр Гаусса
   (d) Линейный усредняющий фильтр (box blur)
   (e) Фильтр Собеля

   Если методы имеют входные параметры, необходимо иметь возможность изменять их пользователю.

8. Реализуйте описанный в статье (тык на ссылку) алгоритм увеличения изображения. При реализации алгоритма можно использовать как методы численного интегрирования, которые вы реализовали в прошлой работе, так и встроенные алгоритмы.

9. Дополнительно. Изобразите фурье-спектр изображения. Попробуйте применить различные преобразования к полученному фурье-спектру.

10. Дополнительно. Реализовать высокочастотную и низкочастотную фильтрацию изображения. Путем удаления низкочастотной составляющей, получить изображение с повышенной резкостью, которое содержит края.

## Ограничения и требования

1. Работа выполняется на языке python 3.11+

2. Среди библиотек, которые разрешено использовать: numpy, cv2, PIL, matplotlib или plotly.

3. Использование встроенных методов фильтрации, обработки изображений, а также алгоритмов, которые требуется реализовать, снижает максимальный балл за работу до 50% баллов от максимума.

## Полезные ссылки

1. Курс CSC по обработке изображений (2013 год)

2. Курс CSC по обработке изображений (2019 год)

3. О свертке простыми словами от Yadro

4. Компьютерная обработка изображений (НОЦ Фотоники ИТМО)

5. Применение ряда Фурье для аппроксимации изображений