Реализация второго уровня прямого и обратного вейвлет-преобразования Хаара

**Реализация второго уровня прямого преобразования**

Провёл небольшой рефакторинг кода предыдущей версии программы, создал вспомогательные функции, которые выполняют чтение / запись в файлы не через статически указанные в программе их пути, а в качестве параметров функции, чтобы удобнее было реализовывать новую логику, т.к. появляется много новых файлов и компонентов.

Далее я поделил исходные директории с компонентами и текстовыми данными на следующие две:

1. 4\_components\_1 и txt\_data\_1 – директории с данными о первом уровне вейвлет-преобразования, имеют аналогичную структуру с директориями 4\_components и txt\_data, которые были описаны в предыдущем отчёте
2. 4\_components\_2 и txt\_data\_2 – здесь уже хранятся данные о втором уровне вейвлет-преобразования. В каждой из них есть подкатегории LL/, LH/, HL/ и HH/. В каждой из них уже хранятся данные, полученные после применения вейвлет преобразования к каждому из компонентов, полученных после первого запуска алгоритма.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Результаты после двухуровневой обработки**

В результате после обработки я получил 20 компонентов-изображений. 4 для исходного файла и по 4 для каждого из 4-х полученных из исходного файла.

Их можно найти по указанному выше на скринах пути.

**Реализация второго уровня обратного преобразования**

Для восстановления исходного изображения так же, как и в предыдущей версии программы, я применял функцию к компонентам, полученным после первого преобразования Хаара.

Для восстановления каждого из компонентов первого уровня преобразования, я применял алгоритм к компонентам из соответствующей для каждого директории второго преобразования, т.е. для компонентов из LL/, LH/, HL/ и HH/ соответственно.

В результате получились следующие файлы:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

1-й файл – восстановленное исходное изображение

Следующие 4 – восстановленные компоненты первого прямого преобразования.

С исходными совпадают.

**Работа с программой**

Теперь при запуске программы можно выбрать не только тип преобразования (прямое / обратное), но и уровень разложения для каждого (пока что только 1 или 2). Далее в соответствии с выбором программа выполняет свою функцию.

Единственное, отсутствует проверка на то, что если, к примеру, изначально изображение разложили на 1 уровень прямым преобразованием, а для обратного преобразования запросили 2 уровня, данную ошибку программа не обработает, нужно отслеживать самостоятельно, но если необходимо, то могу и эту проверку без проблем реализовать.

Скорее всего, программа либо выдаст ошибку (если нет ранее сохранённых компонентов для 2 уровня), либо просто неверный результат.