

Министерство образования и науки РФ
Федеральное государственное автономное образовательное
учреждения высшего образования
«Национальный исследовательский технологический университет «МИСИС»
Кафедра инженерной кибернетики

**Отчет по лабораторной №1 на тему:
«Форматы растров и гамма-коррекция»**

по дисциплине:
«Компьютерное зрение»
Направление подготовки: 01.03.04 Прикладная математика

Семестр 5

Выполнил:
Фейгенберг М.И.

БПМ-23-ПО-3

30.11.2025

Подпись: _____

Проверил:
Полевой Д.В.

(оценка)

(дата проверки)

Подпись: _____

Москва 2025

Цель

создание функционала описания и идентификации параметров растровых изображений. Задача включает автоматизацию проверки соответствия физических параметров файлов их именам и исследование влияния гамма-коррекции на визуальное восприятие градиентных изображений.

Задание

- Реализация функции получения строкового дескриптора изображения вида WWWxNNNN.C.TYPE.
- Реализация механизма чтения списков файлов из .lst конфигураций с использованием относительных путей.
- Генерирование тестовых изображений и функции гамма-коррекции для 8-битных одноканальных изображений.

Инструкции по сборке и запуску

Весь код используемый и описываемый в данной лабораторной работы можно найти по ссылке:

(<https://github.com/Asbecov/misis2025f-23-03-feigenberg-m-i/tree/master>)

Для сборки исполняемых файлов проекта используется система CMake, необходимо прописать следующее:

```
cmake .
```

```
cmake --build ./build
```

На выходе компиляции получается два исполняемых файлы:

- task01-01 — проверка соответствия имён файлов формату растра.
Пример использования:

```
task01-01 ./task01.lst
```

Вывод - по одной строке на файл:

```
/abs/path/img_1.png      good
```

```
/abs/path/img_2.tiff      bad, should be 0768x0432.3.uint08
```

- task01-02 - гамма-коррекция gray_bars_8u_768x30 изображения. Пример использования:

task01-02 ./collage.png

Вывод - изображение ранее приведенное в лабораторной работе (рис. 2).

Ход работы:

Был реализован функционал описания характеристик изображения:

```
std::string strid_from_mat(const cv::Mat& img, const int n = 4);
```

Функция формирует строку-описатель изображения по шаблону WWWxNNNN.C.TYPE, где:

- WWW - ширина (n цифр с ведущими нулями);
- NNNN - высота (n цифр);
- C - количество каналов (кодируемых цветов);
- TYPE - тип данных в канале (uint08, sint08, uint16, sint16, sint32, real32, real64).

Данные характеристики полностью получаются из модели cv::Mat.

Для изучения характеристик изображений предварительно сгенерировано тестовые файлы с разными характеристиками, создан файл task01.lst, содержащий список имен тестовых изображений.

Для чтения данных файлов реализована функция:

```
std::vector<std::filesystem::path> get_list_of_file_paths(const  
std::filesystem::path& path_lst);
```

Функция использует стандартные потоки ввода-вывода C++ (std::ifstream) и методы std::filesystem для получения абсолютных путей к изображениям.

Консольное приложение task01-01 принимает путь к task01.lst, читает изображения и проверяет соответствие их формата строке-описателю. Результат выводится построчно в формате:

filename.ext good

filename.ext bad, should be 0512x0512.1.uint08

Для изменения и корректировки яркости финального отображения на мониторе пользователя используется нелинейная зависимость между входным и выходным значениями пикселя:

$$V_{out} = 255 \cdot \left(\frac{V_{in}}{255} \right)^{1/\gamma}$$

Рис. 1, формула гамма коррекции используемая в лабораторной.

Где:

V_{in} - исходное значение яркости $[0, 255]$;

gamma - коэффициент коррекции;

V_{out} - результирующее значение.

Опишем функцию для реализации гамма коррекции изображения:

```
cv::Mat gamma_correct(const cv::Mat& img, const double gamma);
```

Для повышения производительности алгоритм реализован через таблицу поиска (Lookup Table — LUT), что позволяет избежать вычисления степени для каждого пикселя изображения индивидуально.

Чтобы применить гамма коррекцию и увидеть ее влияние на входящее изображение опишем функцию генерации тестовых изображений:

```
cv::Mat gen_gray_bars(int width = 768, int height = 30, int stripe_w = 3);
```

Для генерации вертикальных полос, чья ширина задается при вызове функции, используется встроенная функция рисования OpenCv `cv::rectangle`.

Для визуальной инспекции влияния гамма коррекции на изображение, консольное приложение task01-02 сшивает оригиналы и прошедшие через гамма коррекцию изображения в единый коллаж.



Рис. 2, сравнение параметров гамма коррекции.

На рис. 2 можно наблюдать гамма коррекцию при следующих значениях гаммы: {1.8, 2.0, 2.2, 2.4, 2.6}.

Как мы видим, чем выше значение гаммы, тем светлее изображение.

В ходе лабораторной работы:

- освоены методы работы с библиотекой OpenCV;
- создано минимальное дерево сборки CMake;
- реализованы функции анализа изображений, чтения списков и гамма-преобразования;
- изучено использование Look-Up Table (LUT) для ускорения нелинейных преобразований;
- проект структурирован для дальнейшего расширения.