МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ «МИСИС»

Институт информационных технологий и компьютерных наук
Кафедра инженерной кибернетики

Курсовая работа

по дисциплине

«Объектно-ориентированное программирование»

на тему

«Библиотека бинаризации изображений»

Выполнил:
студент 1-го курса
гр. БПМ-22-1 Шестаков Н. А.

Проверил:
доцент, к. т. н. Полевой Д.В.

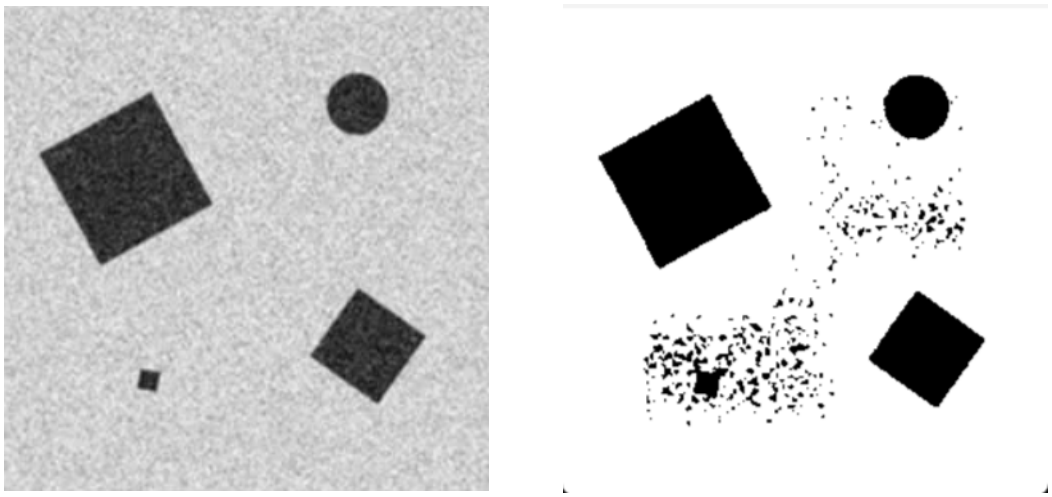
Москва, 2023

Содержание

1 Описание задачи

Целью курсовой работы является создание консольного приложения (CUI) и библиотеки, выполняющей бинаризацию изображения по методу Ниблэка, которые помогут разработчиком в обработке изображений.

Программа предназначена для конечного пользователя.



(a) (b)

Рисунок 1.1 — Исходное и отбинаризованное изображения соответственно.

В основные функции CUI-приложения входят:

— Выполнение процесса бинаризации;

— Выполнение демонстрации алгоритма.

В основные функции библиотеки входят:

— Функция обработки изображения из цветного в черно-белое по методу Ниблэка;

— Функция демонстрации алгоритма;

— Функция генерации *.tex-файла для визуализации алгоритма.

Требования к технической реализации:

— Язык программирования C++, версия не ниже 18;

— Сборка с использованием CMake, версия не ниже 3.15;

— Реализация библиотеки, используя стороннюю библиотеку OpenCV

— Поддержка сборки под разные системы;

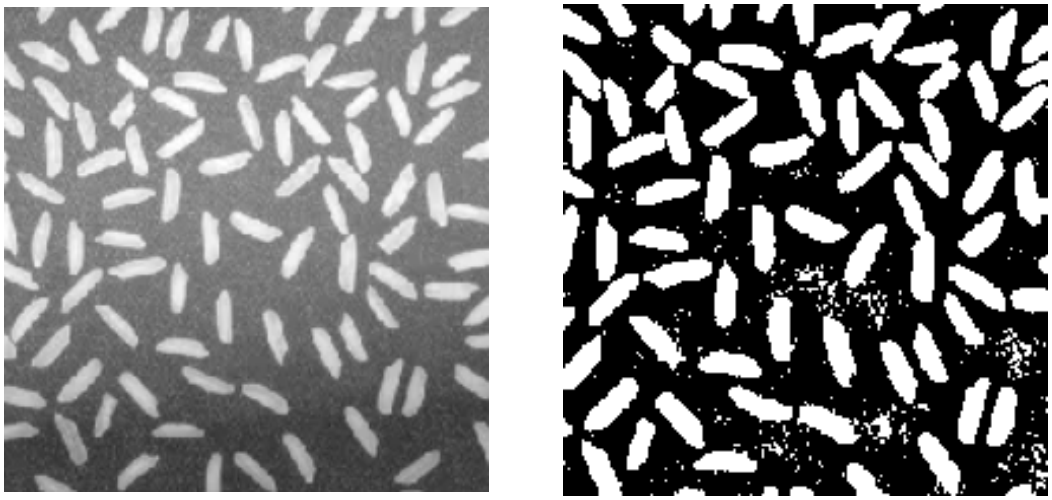
— Реализация взаимодействия с консолью;

— Информативный вывод ошибок в консоль.

2 Пользовательское описание

2.1 Обзор библиотеки

Библиотека NiblackBinarization предоставляет функционал для бинаризации изображений с использованием ряда параметров в зависимости от конечных целей. Так, обязательными из них являются путь до изображения и путь, куда изображение будет помещено. Опциональной командой является демонастрация исходного и отбинаризованного изображения или генерация LaTeX-графика с визуалиазацией хода работы алгоритма.



(a) (b)

Рисунок 2.1 — Исходное и отбинаризованное изображения соответственно.

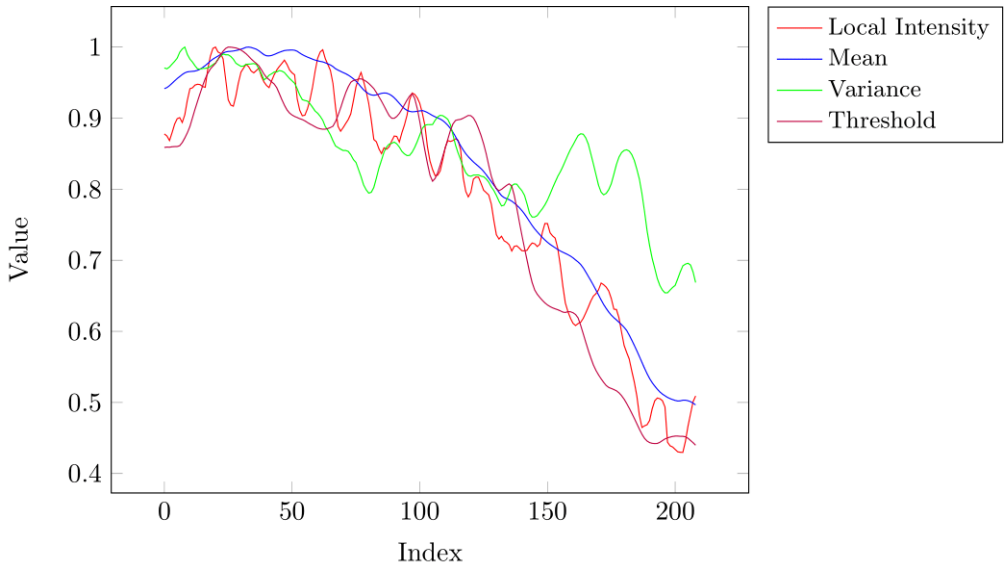


Рисунок 2.2 — Визуализация хода работы алгоритма для 10 строки исходного изображения (см. Рисунок 2.1-а)

2.2 Пользовательская документация

Объявления и описания членов классов находятся в файлах:

— include/niblack/niblack.hpp;

— niblack.cpp.

Класс NiblackBinarization

NiblackBinarization()

Конструктор с параметрами:

```
NiblackBinarization(
        cv::Mat & src,
        const int window_size,
        const double k )
```

Аргументы:

src	матрица, для которой будет применен алгоритм
window_size	размер окна, в котором будет происходить бинаризация
k	коэффициент, влияющий на величину порогового значения

Методы

check_the_image()
bool NiblackBinarization::check_the_image(
cv::Mat image,
)

Метод, который проверяет корректность формата изображения

Аргументы:

image	входное изображение на проверку

Возвращает:
true/false:

VisualisationNiblack()
void NiblackBinarization::demonstrateNiblack(
const cv::Mat & src,
int window_size,
double k,
int selected_row,
std::string output_path)

Метод, осуществляющий визуализацию алгоритма

Аргументы:

src	входное изображение
window_size	размер области, в котором будет осуществляться бинаризация
k	коэффициент, влияющий на величину порогового значения
selected_row	ряд, для которого будет осуществляться визуализация
output_path	путь, куда будет сохранено отбинаризованное изображение и график plot.tex с параметрами алгоритма

drawGraph()

void NiblackBinarization::drawGraph(
std::ofstream & file,
const std::vector< double > & values,
const std::string & color,
int selected_row,
const std::string & label)

Метод, сохраняющий координаты точек в .tex-файл для визуализации

Аргументы:

file	файл для записи координат
values	массив координат точек
color	цвет линии на графике
label	надпись в легенде графика

niblackThreshold()

cv::Mat NiblackBinarization::niblackThreshold(
const cv::Mat & src,
int window_size,
double k,)

Метод, выполняющий бинаризацию изображения по методу Ниблэка

Аргументы:

src	входное изображение
window_size	размер области, в котором будет осуществляться бинариза
k	коэффициент, влияющий на величину порогового значения

Возвращает:
cv::Mat object после бинаризации

plotValues()

void NiblackBinarization::plotValues(
std::filesystem::path filePath,
const std::vector< double > & localIntensity,
const std::vector< double > & meanValues,
const std::vector< double > & varianceValues,
const std::vector< double > & thresholdValues,
int selected_row,)

Метод, в котором формируется структура .tex-файла, а также его содержимое посредством вызова drawGraph-метода

Аргументы:

filePath	путь к исполняему файлу, рядом с которым будет сгенерирован .tex-файл
localIntensity	массив векторов со значениями локальной интенсивност
meanValues	массив векторов со значениями средней яркости
varianceValues	массив векторов со значениями среднего отклонения
thresholdValues	массив векторов порогового значения
selectedRow	номер выбранной для визуализации линии

saveTheImage()

void NiblackBinarization::saveTheImage(
std::filesystem::path filePath,
std::string output_path,
std::string file_name,)

Метод, сохраняющий изображение в формате file_name.png по пути output_path

Аргументы:

src	изображение для сохранения в файл
output_path	путь, куда будет сохранено изображение
file_name	название сохраняемого изображения

setInputImage()
void NiblackBinarization::setInputImage(
const cv::Mat & image,)

Сеттер для image.

Аргументы:

image	входное изображение

setK()
void NiblackBinarization::setK(
double k,)

Сеттер для k.

Аргументы:

k	коэффициент, влияющий на величину порогового значения

setTargetRow()
void NiblackBinarization::setTargetRow(
int target_row,)

Сеттер для target_row.

Аргументы:

target_row	ряд, для которого будет осуществляться визуализация

setWindowSize()
void NiblackBinarization::setWindowSize(
int window_size,)

Сеттер для window_size.

Аргументы:

window_size	размер области, в котором будет осуществляться бинариза

3 Техническое описание

Бинаризация — это процесс преобразования изображения в двухцветное (черно-белое) изображение, где каждый пиксель принимает значение либо 0 (черный), либо 1 (белый). Цель бинаризации заключается в разделении объектов интереса от фона изображения.

Бинаризация основывается на применении порогового значения к интенсивности пикселей. Если интенсивность пикселя выше порогового значения, он классифицируется как объект интереса (белый), в противном случае — как фон (черный).

Метод Ниблэка является одним из алгоритмов бинаризации, который автоматически определяет пороговое значение для каждого пикселя на основе локальных статистических характеристик в его окрестности.

Для каждого пикселя (x, y) метод Ниблэка использует окно фиксированного размера w x w (обычно 15 x 15 или 31 x 31), расположенное вокруг этого пикселя. Затем вычисляются две характеристики внутри окна: среднее значение **(mean)** и стандартное отклонение **(standard deviation)** интенсивностей пикселей в окне.

Пороговое значение p вычисляется по формуле:

$$p = m + st * k \tag{3.1}$$

Где m - среднее значение интенсивностей пикселей в окне, st - стандартное отклонение интенсивностей пикселей в окне, k - пользовательский коэффициент (обычно в диапазоне от -0.5 до 0.5)

Таким образом, наш алгоритм сводится к следующим шагам:

— Загрузка и предварительная обработка изображения: исходное изображение загружается, и выполняются предварительная проверка формата;

— Разбиение изображения на локальные окна: изображение разбивается на неперекрывающиеся локальные окна фиксированного размера. Каждое окно будет использоваться для вычисления локального порогового значения;

— Вычисление порогового значения для каждого окна: для каждого окна выполняется вычисление порогового значения на основе заданной

формулы, которая учитывает среднее значение пикселей окна и некоторую меру локального стандартного отклонения;

— Бинаризация изображения: каждый пиксель изображения преобразуется в черный или белый в зависимости от значения порога, рассчитанного для соответствующего окна, в котором находится данный пиксель;

— Визуализация для строки и сохранение графика интенсивностей в *.tex-файл;

— Сохранение изображений.

4 Инструкция по установке и тестированию

Установка:

Для работы библиотеки NiblackBinarization, генерации документации, демонстрационных и тестовых приложений необходима установка с помощью vcpkg, библиотек OpenCV, а также установка системы генерации Doxygen и дистрибутива TeX.

Для Linux/MacOS:

а) Скачайте исходный код библиотеки и демонстрационного приложения из репозитория проекта по ссылке `https://github.com/DR-Felix/MISIS_OOP_2_sem.git`;

б) Откройте терминал или командную строку **от имени администратора** и перейдите в папку shestakov_n_a;

в) Выполните команду *cmake -DCMAKE_TOOLCHAIN_FILE=[path] -DCMAKE_INSTALL_PREFIX=[path2] -B ./build* (после cmake, [path], [path2] и -B стоят пробелы), где [path] - путь до менеджера библиотек, например, vcpkg, а [path2] – путь, куда будет установлена библиотека/программа.

г) Перейдите в директорию build, выполнив команду: *cd ./build*

д) Соберите cmake, выполнив команду: *cmake --build .*

е) Проинсталлируйте проект, выполнив команду: *cmake --install .*

Для Windows:

а) Скачайте исходный код библиотеки и демонстрационного приложения из репозитория проекта по ссылке `https://github.com/DR-Felix/MISIS_OOP_2_sem.git`;

б) Откройте терминал или командную строку **от имени администратора** и перейдите в папку shestakov_n_a;

в) Выполните команду *cmake -DCMAKE_TOOLCHAIN_FILE=[path] -DCMAKE_INSTALL_PREFIX=[path2] -B ./build* (после cmake, [path], [path2] и -B стоят пробелы), где [path] - путь до менеджера библиотек, например, vcpkg, а [path2] – путь, куда будет установлена библиотека/программа.

г) Перейдите в директорию build, выполнив команду: *cd ./build*

д) Проинсталлируйте проект, выполнив команду: *cmake --build . --target install*

После успешной сборки проекта вы можете запустить тестовое приложение в папке bin, посмотреть LaTeX-документацию в папке docs, которая появилась на одном уровне с bin, в папке niblack будет на-

ходиться готовая библиотека niblack.lib, а в папке lib будет лежать niblackConfig.cmake-файл.

Тестирование:

Взаимодействие с консольным приложением происходит с помощью команд, полный список которых можно получить, используя ключ *niblack_example.exe -h*, где niblack_example.exe - исполняемый файл программы.

Список ключей:

а) -M : Сохранить результат бинаризации в директорию по умолчанию (./result/)

б) -D : Вывести исходное и отбинаризованное изображение в двух окнах

в) -V : Визуализировать параметры алгоритма в graph.tex

г) -w : Ввести параметр-значение размера окна бинаризации (должен быть нечетным, как правило, равен 31)

д) -k : Ввести параметр-значение коэффициента бинаризации (как правило, значение варьируется от -0.5 до 0.5)

е) -t : Ввести параметр-значение ряда, для которого будет выполнена визуализация

ж) -l : Указать путь к изображению

з) -s : Указать путь для сохранения изображения

и) -h : Показать возможные команды

Важное примечание: Команды -l, -s, -w, -k, -t нужно вводить в формате -[command]=[way], т.е. слитно (а в случае с -l и -s без последнего слеша в пути!). **Например**, -k=0.2 или -l=.\images\test2.png

Инструкция по тестированию библиотеки:

а) Откройте папку bin с исполняемым файлом niblack_example.exe;

б) Откройте командную строку **от имени администратора** и введите нужные вам параметры через консольное приложение. Все команды вводятся через пробел.

Внимание: Для выполнения команд -M, -D и -V (то есть сохранения отбинаризованного изображения, демонстрации исходного и отбинаризованного изображения и визуализации параметров алгоритма необходимо ввести путь до исходного изображения и путь, где будет создана папка results с результатами работы алгоритма).

Например, команда

```
.\niblack_example.exe -l=.\images\test2.png -D
```

запустит исполняемый файл niblack_example.exe для изображения, находящегося по пути .\images\test2.png (поскольку папка images с тестовыми изображениями находится в папке bin, можем использовать относительный путь), а с помощью команды -D произойдет демонстрация исходного и отбинаризованного изображения в двух окнах (см. Рисунок 4.1 ниже).

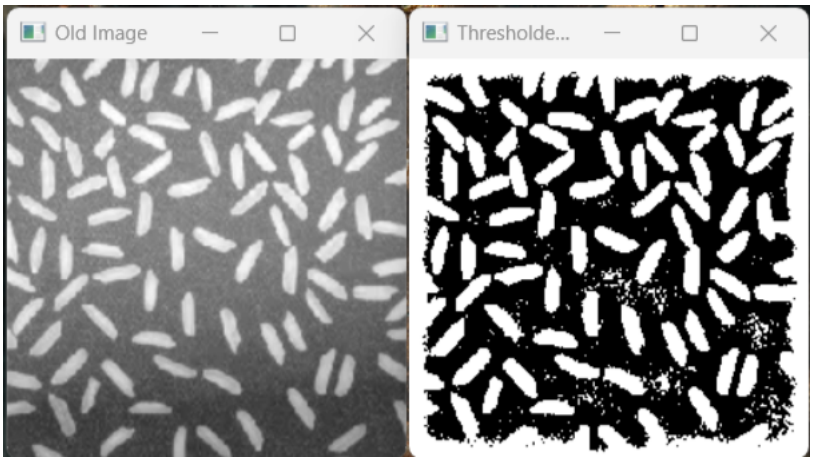


Рисунок 4.1 — Демонстрация результатов работы алгоритма

Примечание: в отсутствие вручную заданных параметров с помощью команд -w, -k и -t, алгоритм запустится с параметрами по умолчанию, равными 31, -0.2 и 10 соответственно.

Использованная литература

а) *Wilhelm Burger, Mark J. Burge* Digital Image Processing. - 3-е изд. - Spinger

б) *Samorodov* A. Fast implementation of the Niblack binarization algorithm for microscope image segmentation // ResearchGate.net. - 2016

в) OpenCV modules // OpenCV URL: https://docs.opencv.org/4.x/ (дата обращения: 2023).