

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

Кафедра комп'ютерних наук

**Теорія розпізнавання образів та класифікації в системах
штучного інтелекту**

Лабораторна робота №7

Виконав:

Студент групи КН-20002Б

Кропивка Анатолій Анатолійович

Київ 2023

Тема: Розпізнавання зашумлених символів.

Мета: Дослідження різних за видом і кількістю використовуваних ознак методів розпізнавання зображень символів, представлених у растровій формі, за різного ступення їх зашумлення.

Підготовка до роботи: Вивчити й уявити призначення і зміст завдання до лабораторної роботи.

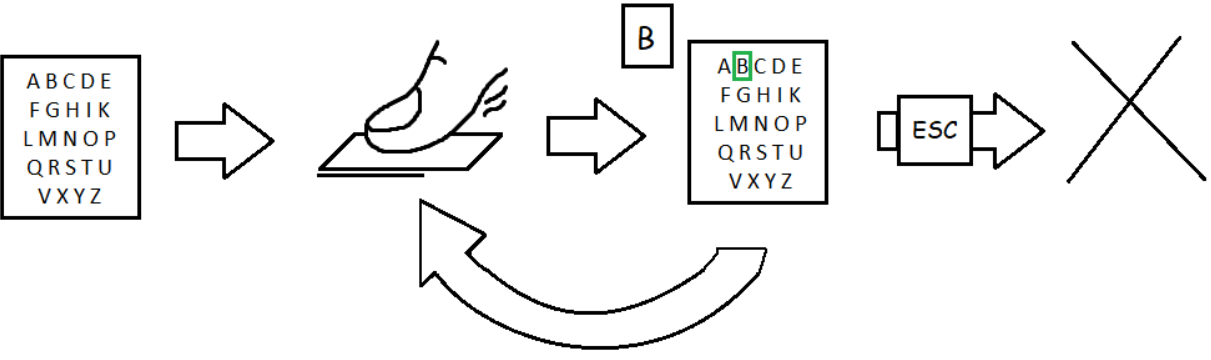
Хід роботи:

1. Ознайомитись з методичною розробкою до лабораторної роботи.
2. Ознайомитись з рекомендованою літературою.
3. Розробити для проведення довідження програмний застосунок.
4. Дослідити зображення сформованих еталонних символів, їх характеристики.
5. Сформувати кілька нових еталонних символів, з різними рівнями та типами відмінностей у відповідності до порядку виконання лабораторної роботи.
6. Сформувати текст, куди входять однакові символи.
7. Вибрати величину зашумлення та метод розпізнавання, зафіксувати результат розпізнавання та відстань до найближчого еталона для кожного символу, що розпізнається.
8. Виконати дослідження у відповідності до підпунктів 3, 4, 5 порядку виконання роботи.
9. Виконання завдання до лабораторної роботи.
10. За результатами досліджень скласти звіт з обґрунтованими висновками.

Опис програми

Програма виконує розпізнавання літер на основі поданої абетки. Робота програми включає наступні кроки:

- Зчитування початкової абетки зображень з файлу *"alphabet.jpg"* та відображення її на екрані.
- В циклі програма очікує введення користувачем великої літери з клавіатури. Якщо натиснута клавіша *esc* (код 27), то програма завершується.
- Перевірка правильності введеної клавіші: допускаються лише великі літери англійського алфавіту (від 'A' до 'Z'). Якщо введена клавіша не задовольняє цю умову, виводиться повідомлення про неправильне натискання клавіші, і програма повертається до очікування введення наступної клавіші.
- Зчитування абетки зображень з файлу *"alphabet.jpg"* та зображення вибраної літери з файлу *request_letter + ".jpg"*. Для кожного зображення також створюється тимчасова копія.
- Перетворення кольорових зображень абетки та вибраної літери в чорно-білі за допомогою конвертації з *COLOR_BGR2GRAY*.
- Розмиття абетки за Гаусом та використання порогового значення для видалення шумів.
- Використання алгоритму *Canny* для виявлення контурів на абетці та вибраній літері.
- Застосування методу *matchTemplate* для пошуку схожості між вибраною літерою та літерами абетки.
- Знаходження позиції літери в абетці за допомогою методу *minMaxLoc*.
- Малювання прямокутника, який обведе вибрану літеру на абетці.
- Відображення абетки та результату розпізнавання на екрані за допомогою функції *imshow*. Додатково відображаються тимчасові копії абетки та вибраної літери.
- Продовження циклу для очікування введення наступної клавіші та повторення вищезазначених кроків.
- При натисканні клавіші *esc* програма завершується, а всі вікна зображень закриваються.



Код програми

```
#include <cstdlib>

#include <iostream>

#include <opencv2/opencv.hpp>

#include <string>

// Зчитати картинку з директорії, якщо ні, то завершити програму.
void read_image(cv::Mat& image, std::string& path)
{
    image = cv::imread(path);

    if (image.empty()) {
        std::cout << "Failed to open an image " << path << std::endl;
        exit(1);
    }
}

int main(int, char**)
{
    std::string path { ".././jpg/" };

    // Вивести початкову абетку.
    cv::Mat initial_alphabet {};
    read_image(initial_alphabet, path + "alphabet.jpg");
    cv::imshow("Alphabet", initial_alphabet);

    while (true) {
        // Зчитати літеру з клавіатури
        char request_letter { static_cast<char>(cv::waitKey(0)) };
        if (request_letter == 27) {
            break;
        }
    }
}
```

```
}
```

```
// Дозволяються тільки великі літери
```

```
if (request_letter < 65 || request_letter > 90) {  
    std::cout << "The wrong key pressed!" << std::endl;  
    continue;  
}
```

```
cv::Mat alphabet {};
```

```
read_image(alphabet, path + "alphabet.jpg");
```

```
cv::Mat temp_alphabet {};
```

```
read_image(temp_alphabet, path + "alphabet.jpg");
```

```
cv::Mat letter {};
```

```
read_image(letter, path + request_letter + ".jpg");
```

```
cv::Mat temp_letter {};
```

```
read_image(temp_letter, path + request_letter + ".jpg");
```

```
// Перевести зображення в COLOR_BGR2GRAY колір.
```

```
cv::cvtColor(temp_alphabet, temp_alphabet, cv::COLOR_BGR2GRAY);
```

```
cv::cvtColor(temp_letter, temp_letter, cv::COLOR_BGR2GRAY);
```

```
// Розмити абетку за Гаусом та прибрати шуми.
```

```
cv::GaussianBlur(temp_alphabet, temp_alphabet, cv::Size(5, 5), 0);
```

```
cv::threshold(temp_alphabet, temp_alphabet, 0, 255, cv::THRESH_BINARY + cv::THRESH_OTSU);
```

```
// Обвести зображення контурами на чорному тлі.
```

```
cv::Canny(temp_alphabet, temp_alphabet, 50, 150);
```

```
cv::Canny(temp_letter, temp_letter, 50, 150);
```

```

// Знайти схожість між пошуковою літерою та літерою з абетки.

cv::Mat result {};

cv::matchTemplate(temp_letter, temp_alphabet, result, cv::TM_CCOEFF_NORMED);

// Знайти позицію літери в абетці.

double min_value {}, max_value {};

cv::Point min_loc {}, max_loc {};

cv::minMaxLoc(result, &min_value, &max_value, &min_loc, &max_loc);

int width { letter.cols };
int height { letter.rows };

cv::Point top_left(max_loc.x, max_loc.y);
cv::Point bottom_right(max_loc.x + width, max_loc.y + height);
cv::Scalar color(0, 255, 0);

cv::rectangle(alphabet, top_left, bottom_right, color, 2);

// Вивести абетки на екран, показати результат.

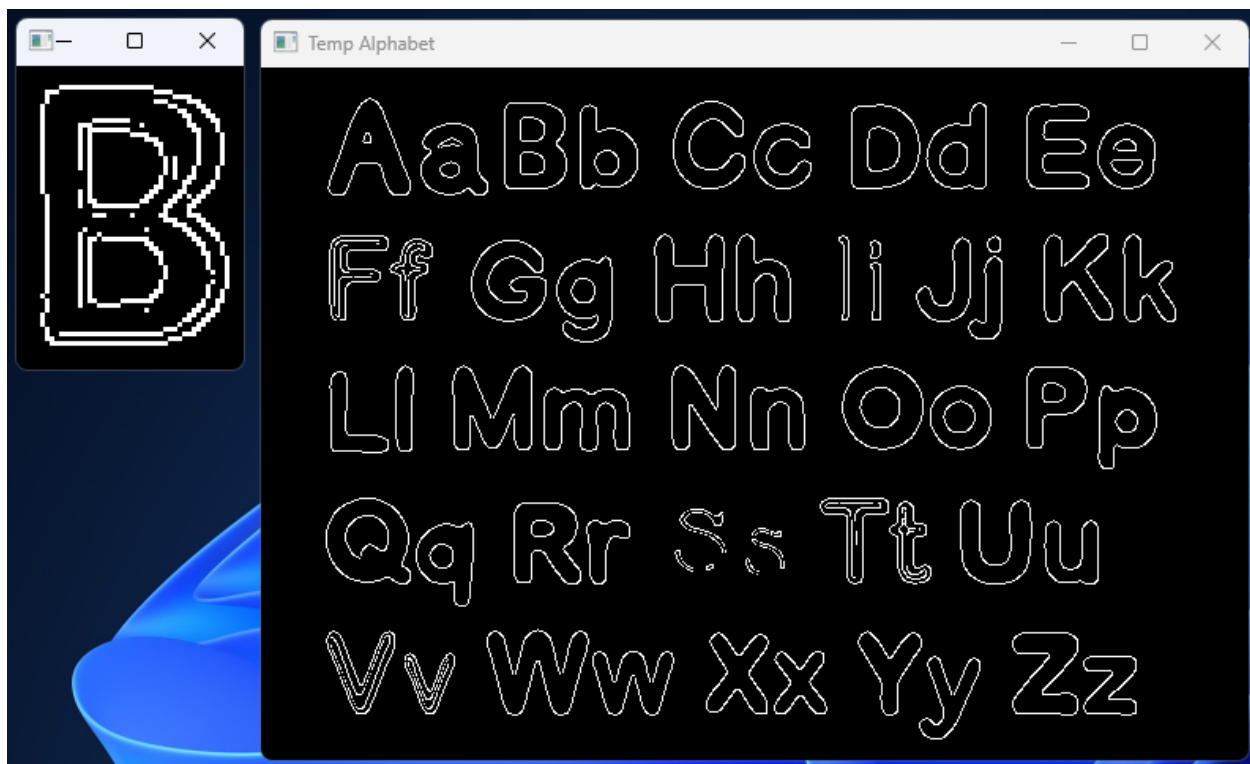
cv::namedWindow("Letter", cv::WINDOW_NORMAL);
cv::resizeWindow("Letter", width * 3, height * 3);
cv::namedWindow("Temp Letter", cv::WINDOW_NORMAL);
cv::resizeWindow("Temp Letter", width * 3, height * 3);

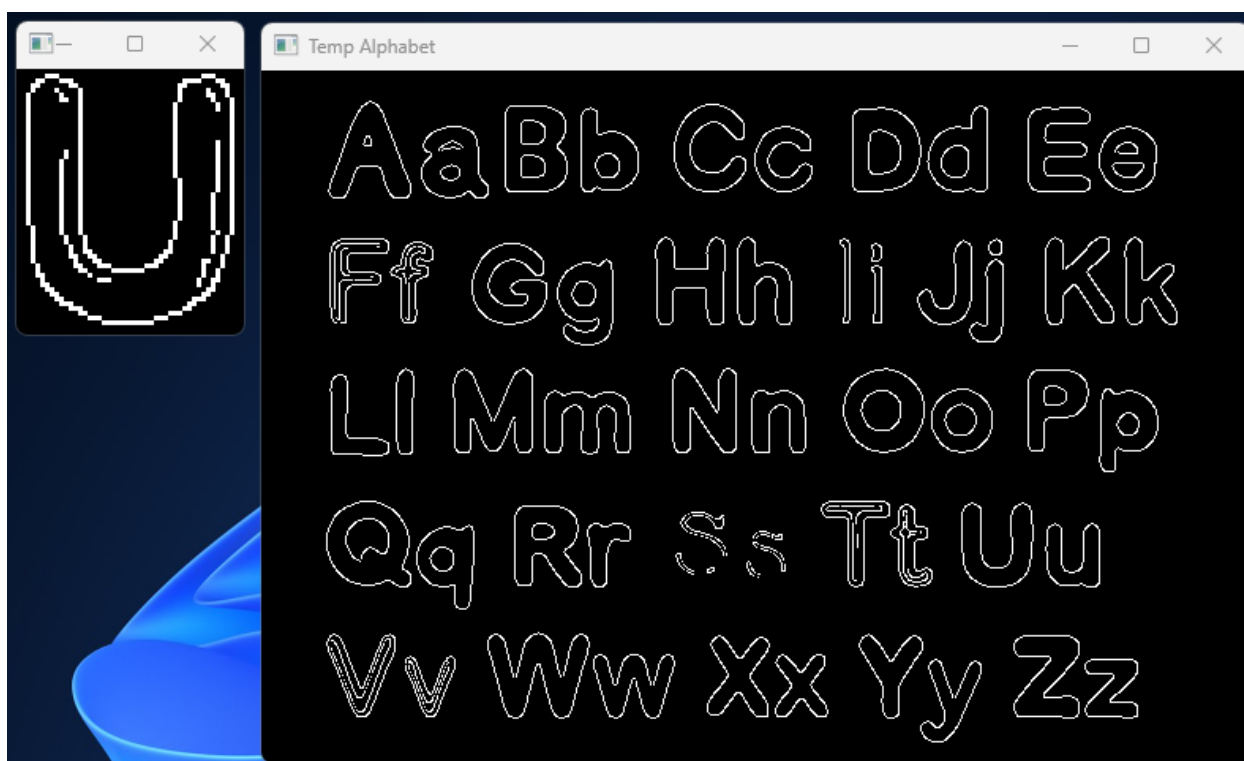
cv::imshow("Letter", letter);
cv::imshow("Alphabet", alphabet);
cv::imshow("Temp Alphabet", temp_alphabet);
cv::imshow("Temp Letter", temp_letter);
}

cv::destroyAllWindows();
return 0;
}

```

Знімки екрану





Висновок: Ця програма демонструє простий приклад розпізнавання літер за допомогою зображень. Вона використовує бібліотеку OpenCV для обробки зображень, зчитування та відображення зображень з файлів. Крім того, програма показує використання функцій для конвертації кольорових зображень в чорно-білі, розмиття, порогової обробки та виявлення контурів зображень. Вона дозволяє користувачеві вводити клавіші для розпізнавання вибраних літер та відображення результатів на екрані. Цей приклад може бути корисним для вивчення основ обробки зображень та розпізнавання образів з використанням OpenCV.