Computer Vision Homework Report

Homework 1

Chinatip Lawansuk

Contents

Chapter 1

Introduction

This project focuses on develop an algorithm to perform principle operations in Computer Vision from scratch.

1.1 Objectives

1. Understand Theory of Computer Vision
2. Improve C++/Python Programming Skill

1.2 Requirements

1. Write a function to convert an image to greyscale image
2. Write a convolution operation with edge detection kernel using zero padding and stride 1
3. Write a pooling operation with using Max pooling, 2×2 kernel, and stride 2
4. Write a binarisation operation (customise the threshold independently).

1.3 System Configuration

1.3.1 Hardware

- CPU: Intel Core-i7
- GPU: NVIDIA
- RAM: 40 GB

1.3.2 Software

- OS: Windows Subsystem Linux x64 (Ubuntu 22.04.3 LTS Kernel Ver. 5.15.90.1)
- GCC Version: `11.4.0 x86_64-linux-gnu`
- OpenCV Version: `4.5.4+dfsg-9ubuntu4`

Chapter 2

Solution, Explanation, and Result

In this project, explanation is embedded in the comment.

2.1 Convert Image to Greyscale Image

Input	:	Matrix of Input Image
Output	:	Matrix of Output Image

```
// offset will be applied for both before, and after the target
   pixel ----
int out_h = ((img.rows + (2 * padding) - kernel.rows) / stride) +
    1;

// create intermediate buffer and add padding, size is w+(2*
   padding),h+(2*padding)
// then copy the content of image to the buffer
Mat padded;
padded = Mat(img.rows + (2 * padding), img.cols + (2 * padding),
   img.type(), Scalar(0, 0, 0));
for (int j = 0; j < img.rows; j++)
{
    for (int i = 0; i < img.cols; i++)
    {
        padded.at<Vec3b>(j+padding,i+padding)=img.at<Vec3b>(j,i);
    }
}
```

Chapter 3

Discussion

Briefly introduce the computer vision tasks you will be performing in this homework.t

3.1 Problem Statements

3.1.1 Problem 1

State the problem statement for Problem 1 here.

3.1.2 Problem 2

State the problem statement for Problem 2 here.kllmm

3.2 Original Images

Figure 3.1: Original Image 1

Figure 3.2: Original Image 2

3.3 Processing Steps and Outputs for Problem 1

3.3.1 Step 1: [Description]

Explanation

Explain what this step does and how it solves or contributes to solving Problem 1.

Output

Figure 3.3: Step 1 Output for Problem 1

3.4 Processing Steps and Outputs for Problem 2

3.5 Code Explanation

```
#include <iostream>
#include <opencv2/opencv.hpp>

int main() {
    // Read the image
    cv::Mat image = cv::imread("original_image1.png");

    // Processing steps for Problem 1
    // ...

    // Processing steps for Problem 2
    // ...

    cv::imwrite("step1_output_problem1.png", image);

    return 0;
}
```

3.6 Conclusion

Summarize what you have learned from completing this homework assignment in the context of computer vision.

Appendix A

Source Code: main.cpp

```
#include <iostream>
#include <opencv2/opencv.hpp>

int main() {
    // Read the image
    cv::Mat image = cv::imread("original_image1.png");

    // Processing steps for Problem 1
    // ...

    // Processing steps for Problem 2
    // ...

    cv::imwrite("step1_output_problem1.png", image);

    return 0;
}
```

Appendix B

Source Code: func.hpp

```
#include <iostream>
#include <opencv2/opencv.hpp>

int main() {
    // Read the image
    cv::Mat image = cv::imread("original_image1.png");

    // Processing steps for Problem 1
    // ...

    // Processing steps for Problem 2
    // ...

    cv::imwrite("step1_output_problem1.png", image);

    return 0;
}
```

Appendix C

Source Code: func.cpp

```
#include <iostream>
#include <opencv2/opencv.hpp>

int main() {
    // Read the image
    cv::Mat image = cv::imread("original_image1.png");

    // Processing steps for Problem 1
    // ...

    // Processing steps for Problem 2
    // ...

    cv::imwrite("step1_output_problem1.png", image);

    return 0;
}
```