摘 要：

在当今数字化的时代，人们对于科技的依赖和需求越来越高。其中，视觉技术作为一种重要的信息处理方式，正得到越来越广泛的应用。在这个背景下，越来越多的人开始关注和研究视觉技术，尤其是视觉算法和应用。视觉技术的应用领域包括但不限于人机交互、智能制造、安防监控、自动驾驶、医疗影像等，而这些领域的应用需求和挑战正不断推动视觉技术的发展和创新。

针对普通用户的视觉技术开发，传统的视觉算法和编程方式可能存在较高的门槛，需要用户具备一定的技术背景和经验。因此，为了使更多的用户能够轻松地使用视觉技术，可视化视觉编程框架应运而生。通过可视化编程的方式，用户可以通过拖拽、连接等简单的操作，实现视觉应用的快速开发，无需编写代码。可视化编程框架不仅降低了用户的学习门槛，还能提高开发效率，节省人工成本，是一种非常具有潜力和前景的技术方向。

本项目旨在以书签智能产线设计与优化为入口，提供一个可视化视觉编程框架，以帮助那些了解少数视觉开发知识但不会编程的普通用户，快速实现视觉应用的开发，而无需编写代码。可视化编程框架是一种编程方法，通过拖拽和放置控件、连接数据以及编写简单的代码，来创建应用程序。

关键词：书签智能产线、可视化编程、 模块化、合格检测、扫码

**Summary**:

In today's digital age, people's dependence and demand for technology are increasing. Among them, visual technology, as an important information processing method, is being increasingly widely applied. In this context, more and more people are paying attention to and researching visual technology, especially visual algorithms and applications. The application fields of visual technology include but are not limited to human-computer interaction, intelligent manufacturing, security monitoring, autonomous driving, medical imaging, etc. The application needs and challenges in these fields are continuously driving the development and innovation of visual technology.

For the development of visual technology for ordinary users, traditional visual algorithms and programming methods may have high barriers, requiring users to have a certain technical background and experience. Therefore, in order to enable more users to easily use visual technology, visual programming frameworks have emerged. Through visual programming, users can achieve rapid development of visual applications through simple operations such as dragging and connecting, without the need to write code. Visual programming frameworks not only lower the learning threshold for users, but also improve development efficiency and save labor costs, making them a highly promising and promising technology direction.

This project aims to provide a visual programming framework with the design and optimization of bookmark intelligent production lines as the entry point, to help ordinary users who have limited knowledge of visual development but do not know how to program, quickly achieve the development of visual applications without the need to write code. Visual programming framework is a programming method that creates applications by dragging and dropping controls, connecting data, and writing simple code.

**Key** **words**: Bookmark intelligent production line, Visual programming, Modularization, Qualification detection, Code scanning

目录

[1 绪论 1](#_Toc177046473)

[**1.1** 课题背景 1](#_Toc177046474)

[**1.2** 国内外研究现状 1](#_Toc177046475)

[**1.3** 本课题任务 2](#_Toc177046476)

[**1.4** 本人任务 3](#_Toc177046477)

[2 图像预处理模块 4](#_Toc177046478)

[**2.1** 需求分析 4](#_Toc177046479)

[**2.2** 图像预处理模块功能统计 4](#_Toc177046480)

[**2.3** 算法设计 4](#_Toc177046481)

[3 扫码模块 5](#_Toc177046482)

[**3.1** 设计目的 5](#_Toc177046483)

[**3.2** 设计实现 5](#_Toc177046484)

[4 次品检测模块 7](#_Toc177046485)

[**4.1** 设计目的 7](#_Toc177046486)

[**4.2** 设计实现 7](#_Toc177046487)

[5 附件 9](#_Toc177046488)

[**6.1 int MatToByte(Mat& inMat, BYTE\* outBYTE)** 9](#_Toc177046489)

[**6.2 int ByteToMat(Mat& outMat, BYTE\* inBYTE, int width, int height, int channel)** 9](#_Toc177046490)

[**6.3 int Threshold(BYTE\* inData,BYTE\* outData,int width, int height, int channel, double thresh, double maxval, int type)** 9](#_Toc177046491)

[**6.4 void Detect(BYTE\* inData, BYTE\* outData, int height\_, int width\_, int channel\_)** 10](#_Toc177046492)

[**6.5 void GarbInit()** 10](#_Toc177046493)

[**6.6 void GarbImg(BYTE\* outData, int width, int height, int channel)** 10](#_Toc177046494)

[**6.7 int MorphologicalOperation(BYTE\* inData, BYTE\* outData, int width, int height, int channel, int kernelShape, int kernelSizeX, int kernelSizeY, int option)** 10](#_Toc177046495)

[**6.8 int Filter(BYTE\* inData, BYTE\* outData, int width, int height, int channel, int level, int option)** 11](#_Toc177046496)

[**6.9 int EdgeDetect(BYTE\* inData, BYTE\* outData, int width, int height, int channel, int low,int high, int option)** 11](#_Toc177046497)

[**6.10 void Defectdetect(BYTE\* inData,int width, int height, int channel,double &score,double &angle)** 12](#_Toc177046498)

[**6.11 double Detect(BYTE\* inData, BYTE\* outData, int width, int height, int channel, char\* codedata)** 12](#_Toc177046499)

# 1 绪论

**1.1** 课题背景

随着科技的快速发展，自动化和智能化生产线在各个行业得到广泛应用。传统的生产方式通常依赖于人工操作，存在着效率低下、成本高昂、容易出错等问题。针对这些问题，智能生产线应运而生，通过引入先进的技术和自动化设备，可以显著提高生产效率、降低人力成本，并提高产品质量和一致性。

在智能书签生产线上，需要使用到机器视觉检测的功能，而传统的文本编程对非专业人士来说可能存在一定的难度，需要学习编程语言的语法和逻辑，这对于那些没有编程经验的人来说可能是一个挑战。

可视化视觉编程框架的出现是为了解决这个问题。它将编程过程可视化，通过图形化界面和直观的交互方式，使非编程专业人士能够更容易地理解和参与到软件开发中。这种框架可以让用户使用图形元素和图形符号来表示编程概念和逻辑，而不是直接编写代码。通过拖放、连接和配置图形元素，用户可以构建程序的逻辑流程，定义变量和函数，以及实现交互效果。

可视化视觉编程框架的研究背景还源于对计算思维和创造力的培养的需求。编程能力已经成为21世纪的核心能力之一，培养学生的计算思维和创造力对他们未来的发展至关重要。可视化视觉编程框架为学生和非专业人士提供了一个友好的编程环境，帮助他们理解抽象的编程概念，培养解决问题和创造新事物的能力。

**1.2** 国内外研究现状

可视化编程框架是一种通过图形化界面来创建程序和应用程序的方法，使非编程专业人士能够更轻松地参与到软件开发中。近年来，可视化编程框架在国内外的研究和应用取得了显著进展。

国外方面，Scratch是麻省理工学院开发的一款面向儿童和初学者的可视化编程框架。在国内，Scratch得到了广泛的应用和研究。一些研究工作关注于Scratch在教育领域的应用，探索如何通过Scratch培养学生的创造力、计算思维和问题解决能力。另外，也有研究致力于对Scratch进行功能扩展和改进，使其适用于更多的应用场景。Blockly是由谷歌开发的一款开源可视化编程框架。在国外，Blockly被广泛用于教育和编程普及。研究人员关注于Blockly在学校教育中的应用，探索如何通过Blockly培养学生的计算思维和编程能力。此外，还有研究工作致力于将Blockly应用于物联网和机器人编程等领域，扩展其应用范围。

国内方面，VPL是中国科学院软件研究所开发的一种面向嵌入式系统的可视化编程框架。VPL提供了一个图形化界面，使非专业人员能够更轻松地进行嵌入式软件开发。它具有易于理解的编程块和图形元素，支持用户进行逻辑控制、传感器控制等操作，以实现物联网设备的开发和控制。CoBlocks是中国的一个开源可视化编程框架，它基于Scratch的图形化编程语言，并扩展了更多的功能和组件。CoBlocks提供了更多用于数学、逻辑和图形绘制等方面的编程块，使学生能够创建更复杂和多样化的项目。CoBlocks被广泛应用于学校教育和创客教育，帮助学生提高计算思维和创造力。

综上可知，国内外在可视化编程框架研究上都有很多成就，然而对于可视化视觉编程框架方面研究较少，因此具有广阔的发展前景。通过进一步的研究和创新，将可视化视觉编程框架与机器视觉技术相结合，提供丰富的组件和算法库，涵盖常见的机器视觉任务，如目标检测、人脸识别、姿态估计等，有望让更多的非专业人士能够轻松地利用机器视觉来解决实际问题，推动机器视觉技术的普及和应用。

**1.3** 本课题任务

本项目需要利用视觉传感技术，实现对书签的准确定位，并将定位信息传递给电控组，以完成书签的抓取操作。这项工作需要视觉处理技术以及TCP/IP通讯技术，并与电控组紧密合作，确保视觉与机械操作的精准协调。

其次，本团队需要实现了对二维码和条码的识别。通过对二维码进行准确识别，判断二维码是否存在错误，并及时进行剔除处理。这项工作的完成为产线的高效运行提供了重要保障。

此外还将所有视觉功能进行了模块化封装。这意味着本团队需要将每个视觉功能独立成模块，使其可以方便地被调用和组合。这种模块化的设计，不仅提高了开发效率，还为后续功能的拓展和维护提供了便利。

同时，本团队需要为初级工程师设计一个操作界面。该界面包含了拖拽视觉模块的功能。通过简单的拖拽操作，工程师可以自由组合视觉功能模块，实现所需的功能。这样的设计使得整个系统对工程师更加友好，让工程师能够快速定制并调整视觉流程。

最后，本团队需要将初级工程师搭建的功能模块进行封装，使其成为一个简单易用的.exe文件，让无任何视觉基础的普通工人也能使用。普通工人只需按照简单的指引操作，即可完成视觉任务，无需具备专业的视觉知识，提高操作的普及性和可用性。

**1.4** 本人任务

本人在该项目中承担了多项任务，其中包括扫码模块和部分图像预处理模块的开发，数据流接口的转换，第二工位检测，以及dll联调等工作。

该项目中包含多种视觉算法和功能，其中重要的一项是扫码识别模块。该模块旨在提供快速、准确的扫码解码功能，简化用户的工作流程，提高使用效率。二维码和条码广泛应用于商业、物流、政务等领域，因此该模块对于提高用户的扫码体验具有重要意义。此外，该模块还可以作为一个基础模块，为其他功能提供基础数据和操作接口，促进整个项目的开发和维护。

图像预处理可以对图像进行预处理和优化，使得后续的算法能够更加准确地识别和分析图像中的目标物体。通常，原始的图像数据往往包含噪声、失真、亮度不均等问题，这些问题会对后续算法的准确性和可靠性造成影响。通过图像预处理，可以对图像进行去噪、去除失真、亮度均衡等操作，从而提高图像的质量和可信度，增强算法的鲁棒性和准确性。因此，在视觉算法的应用中，图像预处理是一个不可或缺的环节，能够有效提高算法的性能和可靠性。

此外，需要使用自己开发的预处理模块和扫码模块进行相机检测任务的重要性和作用不可忽视，这些模块的整合发挥着关键的作用如：增加生产效率、确保信息准确性。自动化相机检测过程可以快速而准确地对书签进行检测和分析，避免了传统人工检查的时间和劳动成本。这样可以加快生产速度，提高产线的整体效率，同时减少了人为错误的可能性。扫码模块的应用能够验证书签上的二维码的正确性，包括内容、格式和有效性。这确保了书签所承载的信息是准确的，消费者可以正确地获取相关信息，提升用户体验和满意度。

相机检测任务中的预处理模块和扫码模块在智能书签生产产线上的重要性和作用是提高质量控制、增加生产效率、确保信息准确性，并提供定制性和灵活性。这些模块的应用使得生产过程更可靠、高效，并确保最终产品的质量和功能符合要求。

在可视化编程中，将数据从Mat格式转换为byte格式非常重要，因为Mat格式可能包含了大量的数据，而byte格式可以更有效地在程序之间传输和存储。此外，Mat格式的数据在传输和存储过程中也可能会发生错误或丢失，而byte格式的数据可以更好地保证数据的完整性和可靠性。因此，将数据从Mat格式转换为byte格式是可视化编程中非常关键的一步，它可以使程序更加稳定和高效。

由于该项目涉及多个模块的协作，每个模块都有可能出现问题，特别是在数据交换和调用过程中。因此，需要进行DLL联调以确保各个模块之间的数据传输和函数调用正确无误。这样可以避免因为一个小问题导致整个系统崩溃的情况发生，保证系统的可靠性和稳定性。

# 2 图像预处理模块

**2.1** 需求分析

基于本次的项目任务要求，要在可视化的软件中嵌入图像模块的设计。首先考虑有哪些图像预处理内容，将相同功能的操作封装到一起，提供接口让用户选择具体的算法。其次考虑图像数据的传递方式，使用BYTE\*传递数据，效率高。

**2.2** 图像预处理模块功能统计

在了解调研的过程中，主要了解到的预处理功能有二值化、滤波操作等，具体内容如下表：

|  |  |  |
| --- | --- | --- |
| 预处理操作 | 操作具体分类 | 函数接口 |
| 数据传递 | Mat类型转BYTE\*  BYTE\*类型转Mat | MatToByte  ByteToMat |
| 二值化 | 二值化不同算法 | Threshold |
| 形态学操作 | 闭运算  开运算  膨胀  膨胀 | MorphologicalOperation |
| 图像滤波 | 高斯滤波  中值滤波 | Filter |
| 边缘检测 | Canny算子  Sobl算子 | EdgeDetect |

表2.2-1 预处理模块接口

**2.3** 算法设计

函数入口使用BYTE\*类型传递，在函数内部使用ByteToMat将图片数据转换为Mat类型，方便使用Opencv处理，最后将处理完的图像使用MatToByte传出，这样在c#端可以接收到处理完的图片数据。

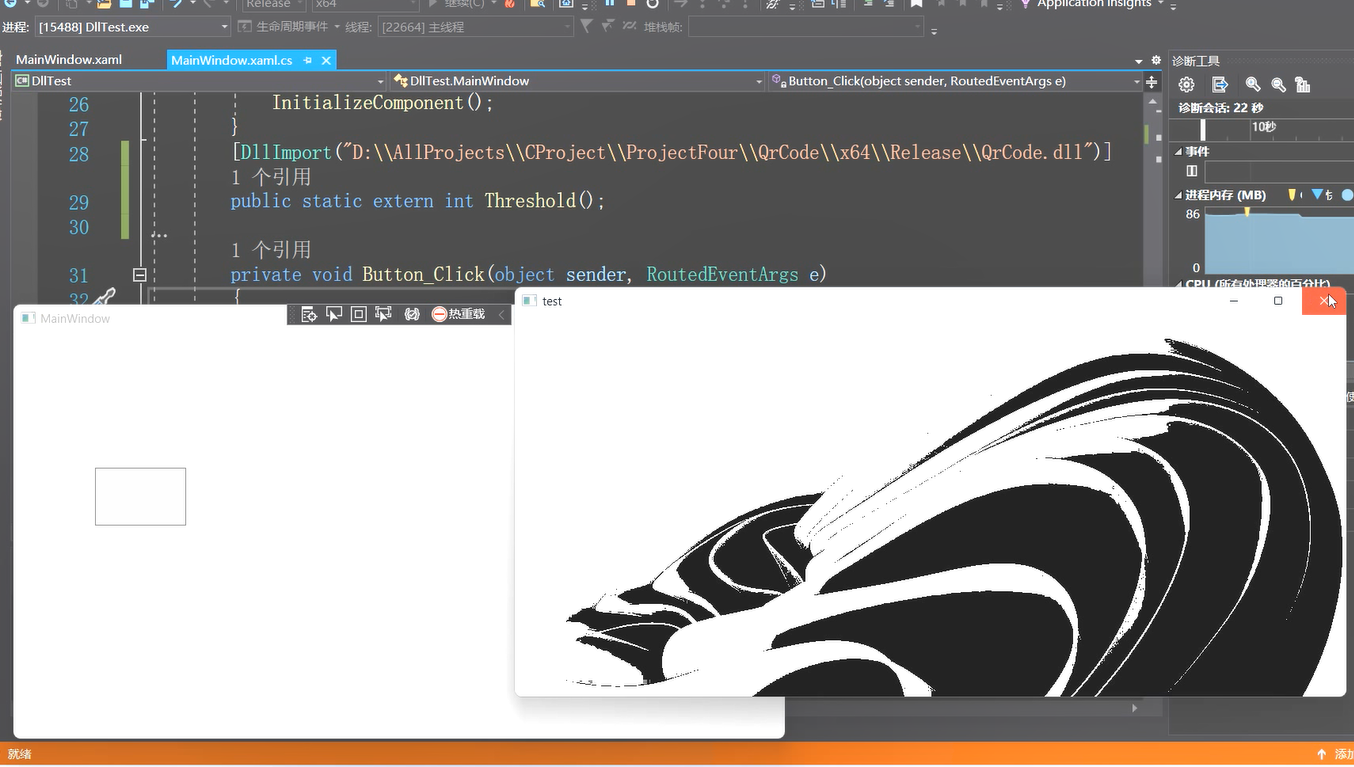


图2.3-1 数据流传递及处理

# 3 扫码模块

**3.1** 设计目的

可视化视觉编程框架中的扫码模块设计的主要目的是为了实现对二维码或条形码的扫描和解码操作。这个模块的设计旨在让用户能够通过简单的操作，使用摄像头或其他视觉传感器来捕捉图像中的二维码或条形码，并将其转换为可识别的文本或数据。

**3.2** 设计实现

在扫码前进行图像预处理，提高扫码成功率；前期使用成熟的扫码库和算法，例如opencv\_wechat和zbar库，避免重复开发和时间浪费；后期同一实现二维码和条码的快速扫描与解码算法，保证高效率和准确性；提供简单易用的接口和文档，方便其他模块的集成和使用；提供可扩展的接口和参数，以满足不同需求的用户。



图3.2-1 扫码测试

在初期，扫码模块主要使用开源扫码解码库实现，使用opencv库函数将图片二值化，使原图留下含码部分，根据长宽比判断码的类型，调用opencv\_wechat函数将图片里的二维码解析出来，调用zbar库函数将条码解析出来，流程如下图所示：



图3.2-2 前期扫码流程图

后期优化算法，使其能够同时解析二维码和条码等多种类型，降低算法复杂度，流程图如下所示：



图3.2-3 后期扫码流程

基于上述流程编写完代码后，需要对该模块的可用性做测试，以确保能够使其正确的用于书签智能产线上。在测试过程中，本人对二维码和条形码分别进行了10次扫描测试，每次测试记录扫描时间和成功率，并记录扫描过程中可能存在的问题，如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试编号** | **扫描类型** | **扫描时间（秒）** | **成功率（%）** | **备注** |
| 1 | 二维码 | 1.2 | 72 | 光线较暗 |
| 2 | 条形码 | 0.3 | 90 | 无异常 |
| 3 | 二维码 | 0.8 | 85 | 码损伤 |
| 4 | 条形码 | 0.2 | 96 | 无异常 |
| 5 | 二维码 | 0.5 | 60 | 光线较亮 |
| 6 | 条形码 | 0.4 | 92 | 无异常 |
| 7 | 二维码 | 1.5 | 70 | 码污损 |
| 8 | 条形码 | 0.2 | 88 | 无异常 |
| 9 | 二维码 | 0.6 | 80 | 光线较暗 |
| 10 | 条形码 | 0.3 | 94 | 无异常 |

表3.2-1 扫码测试

# 4 次品检测模块

**4.1** 设计目的

此智能书签生产线书签次品检测模块的设计旨在提高书签生产线的质量控制能力，确保书签的外观质量和可读性。具体目的包括：检测书签正反面、检测书签图案及图案角度是否正确，以及检测书签上二维码及角度是否正确。

**4.2** 设计实现

为了实现上述目的，本模块采用以下处理流程进行书签次品检测：首先，将书签图像进行二值化处理，将彩色图像转换为黑白图像，以便后续的图像处理操作。接下来，利用图像处理技术，通过寻找轮廓找出书签的位置，确定书签的边界。基于书签的最小外接矩形，进行透视变换，对书签进行校正，以纠正书签的倾斜和变形，提高后续处理的准确性。

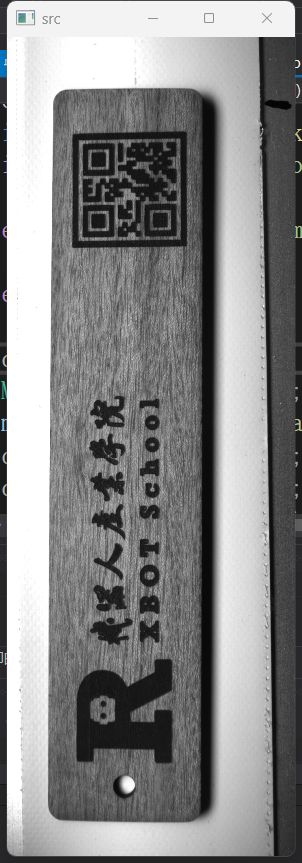


图4.2-1 校正前书签

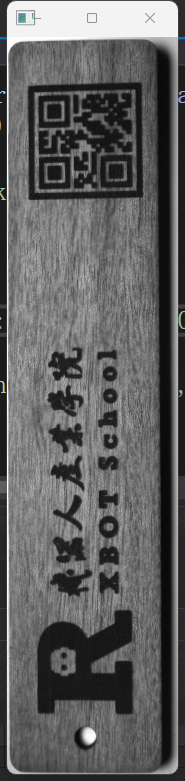


图4.2-2 校正后书签

通过霍夫圆检测算法，对书签上的孔洞或特定形状的标记进行检测，以判断书签的正反面。使用模板匹配算法，逐个角度匹配预定义的图案模板，检测图案是否正确，并记录图案角度的偏差。借助OpenCV微信扫码库，对书签上的二维码进行解码和识别，获取二维码的内容，并检测二维码的角度偏差。

根据上述检测结果，对书签进行分类、标记或剔除次品，将检测结果输出，供后续处理或记录。

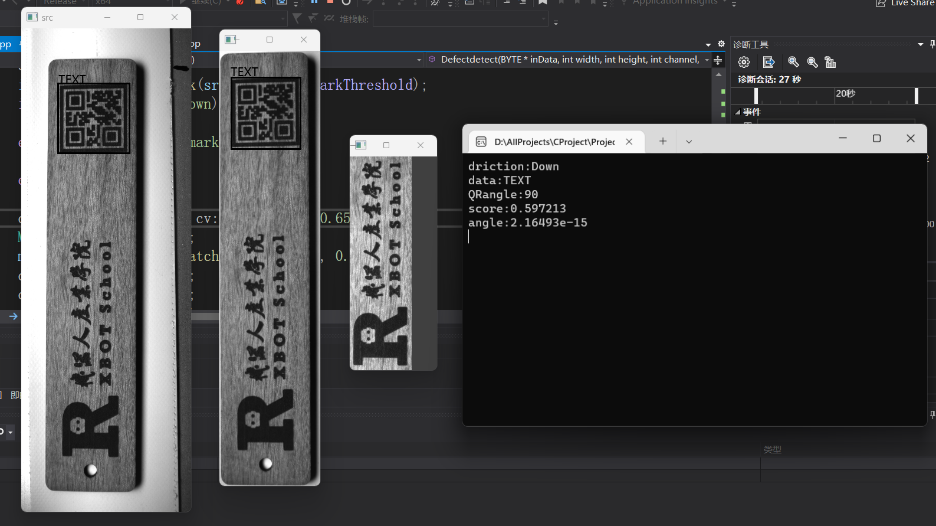


图4.2-3 检测结果

通过以上设计实现，该智能书签生产线书签次品检测模块可以有效地检测书签正反、图案及图案角度、二维码及角度，并对次品进行识别和处理。这将提高书签生产线的质量控制能力，降低次品率，提高生产效率。同时，该模块的设计可以应用于实际生产环境中，为书签生产过程中的质量控制提供可靠的解决方案。

# 5 附件

**6.1 int MatToByte(Mat& inMat, BYTE\* outBYTE)**

函数名称：MatToByte

功能描述：将cv::Mat对象中的像素值转换为BYTE类型数据，存储到指定的内存地址。

参数列表：

inMat: cv::Mat类型，输入图像数据。

outBYTE: BYTE类型指针，输出图像数据存储地址。

返回值：int类型，返回0表示转换成功，返回其他值表示转换失败。

**6.2 int ByteToMat(Mat& outMat, BYTE\* inBYTE, int width, int height, int channel)**

函数名称：ByteToMat

功能描述：将BYTE类型数据转换为cv::Mat对象中的像素值。

参数列表：

outMat: cv::Mat类型，输出图像数据。

inBYTE: BYTE类型指针，输入图像数据存储地址。

width: int类型，图像宽度。

height: int类型，图像高度。

channel: int类型，图像通道数。

返回值：int类型，返回0表示转换成功，返回其他值表示转换失败。

**6.3 int Threshold(BYTE\* inData,BYTE\* outData,int width, int height, int channel, double thresh, double maxval, int type)**

函数名称：Threshold

功能描述：对输入的图像数据进行阈值化处理。

参数列表：

inData: BYTE类型指针，输入图像数据存储地址。

outData: BYTE类型指针，输出图像数据存储地址。

width: int类型，图像宽度。

height: int类型，图像高度。

channel: int类型，图像通道数。

thresh: double类型，阈值。

maxval: double类型，最大值。

type: int类型，阈值化类型。

返回值：int类型，返回0表示处理成功，返回其他值表示处理失败。

**6.4 void Detect(BYTE\* inData, BYTE\* outData, int height\_, int width\_, int channel\_)**

函数名称：Detect

函数说明：用于检测二维码并在图像上标出检测结果。

参数说明：

inData：输入图像数据指针，类型为BYTE\*。

outData：输出图像数据指针，类型为BYTE\*。

height\_：输入图像高度，类型为int。

width\_：输入图像宽度，类型为int。

channel\_：输入图像通道数，类型为int。

返回值：无

**6.5 void GarbInit()**

函数名称：GarbInit

函数说明：用于初始化摄像头。

参数说明：无

返回值：无

**6.6 void GarbImg(BYTE\* outData, int width, int height, int channel)**

函数名称：GarbImg

函数说明：用于从摄像头获取图像数据。

参数说明：

outData：输出图像数据指针，类型为BYTE\*。

width：输出图像宽度，类型为int。

height：输出图像高度，类型为int。

channel：输出图像通道数，类型为int。

返回值：无

**6.7 int MorphologicalOperation(BYTE\* inData, BYTE\* outData, int width, int height, int channel, int kernelShape, int kernelSizeX, int kernelSizeY, int option)**

函数名称：MorphologicalOperation

函数说明：进行形态学操作。

参数说明：

- inData：输入图像数据指针，类型为BYTE\*。

- outData：输出图像数据指针，类型为BYTE\*。

- width：图像宽度，类型为int。

- height：图像高度，类型为int。

- channel：图像通道数，类型为int。

- kernelShape：内核形状（0：矩形，1：十字形，2：椭圆形），类型为int。

- kernelSizeX：内核的 X 方向尺寸，类型为int。

- kernelSizeY：内核的 Y 方向尺寸，类型为int。

- option：形态学操作选项（0：闭，1：开，2：梯度，3：顶帽，4：黑帽，5：膨胀，6：腐蚀），类型为int。

返回值：操作是否成功的标志，类型为int。如果操作成功，则返回0；否则返回其他值。

**6.8 int Filter(BYTE\* inData, BYTE\* outData, int width, int height, int channel, int level, int option)**

函数名称：Filter

函数说明：对图像进行滤波操作。

参数说明：

- inData：输入图像数据指针，类型为BYTE\*。

- outData：输出图像数据指针，类型为BYTE\*。

- width：图像宽度，类型为int。

- height：图像高度，类型为int。

- channel：图像通道数，类型为int。

- level：滤波级别，类型为int。

- option：滤波选项（0：方框滤波，1：均值滤波，2：高斯滤波，3：中值滤波），类型为int。

返回值：操作是否成功的标志，类型为int。如果操作成功，则返回0；否则返回其他值。

**6.9 int EdgeDetect(BYTE\* inData, BYTE\* outData, int width, int height, int channel, int low,int high, int option)**

函数名称：EdgeDetect

函数说明：对图像进行边缘检测。

参数说明：

- inData：输入图像数据指针，类型为BYTE\*。

- outData：输出图像数据指针，类型为BYTE\*。

- width：图像宽度，类型为int。

- height：图像高度，类型为int。

- channel：图像通道数，类型为int。

- low：边缘检测的低阈值，类型为int。取值范围为0-100。

- high：边缘检测的高阈值，类型为int。取值范围为0-100。

- option：边缘检测选项（0：Canny边缘检测，1：Sobel边缘检测，2：Laplacian边缘检测），类型为int。

返回值：操作是否成功的标志，类型为int。如果操作成功，则返回0；否则返回其他值。

**6.10 void Defectdetect(BYTE\* inData,int width, int height, int channel,double &score,double &angle)**

函数名称：Defectdetect

功能描述：检测缺陷并返回得分和角度。

参数列表：

-inData: BYTE 类型指针，输入图像数据存储地址。

-width: int 类型，输入图像的宽度。

-height: int 类型，输入图像的高度。

-channel: int 类型，输入图像的通道数。

-score: double 引用，输出的缺陷得分。

-angle: double 引用，输出的缺陷角度。

返回值：无。

**6.11 double Detect(BYTE\* inData, BYTE\* outData, int width, int height, int channel, char\* codedata)**

函数说明：用于检测书签上的二维码并获取二维码信息及角度。

参数说明：

inData：输入图像数据指针，类型为BYTE\*。

outData：输出图像数据指针，类型为BYTE\*。

width：图像宽度，类型为int。

height：图像高度，类型为int。

channel：图像通道数，类型为int。

codedata：二维码信息存储指针，类型为char\*。

返回值：二维码的角度，类型为double。