# 本科毕业设计（论文）

## 基于 LaTeX 的毕设模板的设计与实现

## The Design and Implementation of Thesis Template Based on LaTeX


| | |
|---|---|
| 学　　院： | 软件学院 |
| 专　　业： | 软件工程 |
| 学生姓名： | 张三 |
| 学　　号： | 18300704 |
| 指导教师： | 刘四 |



北京交通大学

2025 年 5 月

# 学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：　　指导教师签名：

签字日期：　　年　　月　　日　　签字日期：　　年　　月　　日

# 中文摘要

**摘要：**[鼠标左键单击选择该段落，输入替换之。内容为小四号宋体。] 中文摘要应将论文的内容要点简短明了地表达出来，约 400 字左右，字体为宋体小四号。内容应包括工作目的、研究方法、成果和结论。要突出本论文的创新点，语言力求精炼。为了便于文献检索，应在本页下方另起一行注明论文的关键词（3-5 个），如有可能，尽量采用《汉语主题词表》等词表提供的规范词。图 X 幅，表 X 个，参考文献 X 篇。

**关键词：**关键词 1；关键词 2；关键词 3

# ABSTRACT

**ABSTRACT:**

[鼠标左键单击选择该段落，输入替换之。内容为小四号 Times New Roman。] 与中文摘要内容要相对应。“the” quick brown fox jumps over the lazy dog

**KEYWORDS:** KEYWORD1; KEYWORD2; KEYWORD3

# 目　　录

# 1 引言

[鼠标左键单击选择该段落，输入替换之。内容为小四号宋体。] 引言（或绪论）简要说明研究工作的目的、范围、相关领域的前人工作和知识空白、理论基础和分析、研究设想、研究方法和实验设计、预期结果和意义等。应言简意赅，不要与摘要雷同，不要成为摘要的注释。一般教科书中有的知识，在引言中不必赘述。

# 2 【1 级标题，三号黑体字】

[鼠标左键单击选择该段落，输入替换之。内容为小四号宋或楷体字。] 学位论文为了需要反映出作者确已掌握了坚实的基础理论和系统的专门知识，具有开阔的科学视野，对研究方案作了充分论证，因此，有关历史回顾和前人工作的综合评述，以及理论分析等，可以单独成章，用足够的文字叙述。正文是学位论文的核心部分，占主要篇幅，可以包括：调查对象、实验和观测方法、仪器设备、材料原料、实验和观测结果、计算方法和编程原理、数据资料、经过加工整理的图表、形成的论点和导出的结论等。

由于研究工作涉及的学科、选题、研究方法、工作进程、结果表达方式等有很大的差异，对正文内容不能作统一的规定。但是，必须实事求是，客观真切，准确完备，合乎逻辑，层次分明，简练可读。

**图：**

图包括曲线图、构造图、示意图、框图、流程图、记录图、地图、照片等。

图应具有“自明性”。

图应有编号。图的编号由“图”和从“1”开始的阿拉伯数字组成，图较多时，可分章编号。

图宜有图题，图题即图的名称，置于图的编号之后。图的编号和图题应置于图下方。

照片图要求主题和主要显示部分的轮廓鲜明，便于制版。如用放大缩小的复制品，必须清晰，反差适中。照片上应有表示目的物尺寸的标度。

图片示例 1：

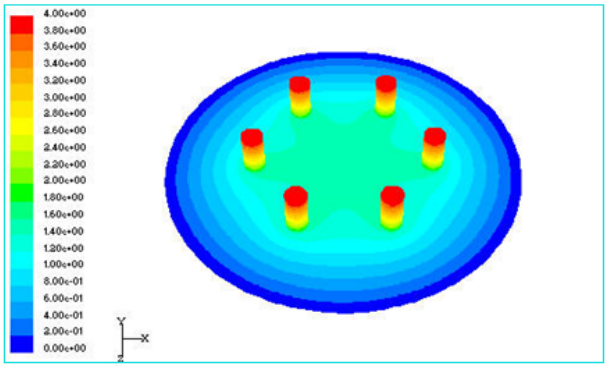



图 2-1 太合金多炭钢铁产品柱扭曲局部受力分析示意图

**表：**

表应具有“自明性”。

表应有编号。表的编号由“表”和从“1”开始的阿拉伯数字组成，表较多时，可分章编号。表较多时，可分章编号。表较多时，可分章编号。表较多时，可分章编号。

表宜有表题，表题即表的名称，置于表的编号之后。表的编号和表题应置于表上方。

表的编排，一般是内容和测试项目由左至右横读，数据依序竖读。

表的编排建议采用国际通行的三线表。

如某个表需要转页接排，在随后的各页上应重复表的编号。编号后跟表题（可省略）和“（续）”，置于表上方。

续表均应重复表头。

表格示例 1：

表 2-1 国际单位制的基本单位

| 量的名称 | 单位名称 | 单位符号 |
|---|---|---|
| 长度 | 米 | m |
| 质量 | 千克 (公斤) | kg |
| 时间 | 秒 | s |
| 电流 | 安 [培] | A |
| 热力学温度 | 开 [尔文] | K |
| 物质的量 | 摩 [尔] | mol |
| 发光强度 | 坎 [德拉] | cd |

**公式：**

论文中的公式应另行起，并缩格书写，与周围文字留足够的空间区分开。

如有两个以上的公式，应用从“1”开始的阿拉伯数字进行编号，并将编号置于括号内。公式的编号右端对齐，公式与编号之间可用“…”连接。公式较多时，可分章编号。

公式示例 1：

$$\phi = \frac{D_p^2}{150}\frac{\psi^3}{(1-\psi)^2} \tag{2-1}$$

$$C_2 = \frac{3.5}{D_p}\frac{(1-\psi)}{\psi^3} \tag{2-2}$$

式中 $D_p$ ——多孔质材料的平均粒子直径 ($m$)；

$\psi$ ——孔隙度（孔隙体积占总体积的百分比）；

$\phi$ ——特征渗透性或固有渗透性，与材料的结构性质有关 ($m^2$)。

较长的公式需要转行时，应尽可能在“=”处回行，或者在“+”、“－”“×”、“/”等记号处回行。

公式中分数线的横线，其长度应等于或略大于分子和分母中较长的一方。

如正文中书写分数，应尽量将其高度降低为一行。如将分数线书写为“/”，将根号改为负指数。

公式示例 2：

将 $\frac{1}{\sqrt{2}}$ 写成 $1/\sqrt{2}$ 或 $2^{-1/2}$

**引文标注**

论文中引用的文献的标注方法遵照 GB/T 7714 — 2005，可采用顺序编码制，也可采用著者—出版年制，但全文必须统一。

**注释**

当论文中的字、词或短语，需要进一步加以说明，而又没有具体的文献来源时，用注释。注释一般在社会科学中用得较多。

应控制论文中的注释数量，不宜过多。

由于论文篇幅较长，建议采用文中编号加“脚注”的方式。最好不用采用文中编号加“尾注”。

## 2.1 【2 级标题，小三号黑体字】

[鼠标左键单击选择该段落，输入替换之。内容为小四号宋或楷体字。]

### 2.1.1 【3 级标题，四号黑体字】

[鼠标左键单击选择该段落，输入替换之。内容为小四号宋或楷体字。]

# 3 【1 级标题，三号黑体字】

## 3.1 【2 级标题，小三号黑体字】

[鼠标左键单击选择该段落，输入替换之。内容为小四号宋或楷体字。]

### 3.1.1 【3 级标题，四号黑体字】

[鼠标左键单击选择该段落，输入替换之。内容为小四号宋或楷体字。]

# 4 【1 级标题，三号黑体字】

## 4.1 【2 级标题，小三号黑体字】

[鼠标左键单击选择该段落，输入替换之。内容为小四号宋或楷体字。]

### 4.1.1 【3 级标题，四号黑体字】

[鼠标左键单击选择该段落，输入替换之。内容为小四号宋或楷体字。]

# 5 结论【1 级标题，三号黑体字】

[鼠标左键单击选择该段落，输入替换之。内容为小四号宋或楷体字。] 论文的结论是最终的、总体的结论，不是正文中各段的小结的简单重复。结论应该准确、完整、明确、精练。如果不可能导出应有的结论，也可以没有结论而进行必要的讨论。可以在结论或讨论中提出建议、研究设想、仪器设备改进意见以及尚待解决的问题等。

# 6 各种测试

## 6.1 图表测试

### 6.1.1 单张图片

单张普通插图，及其引用，如图??所示。

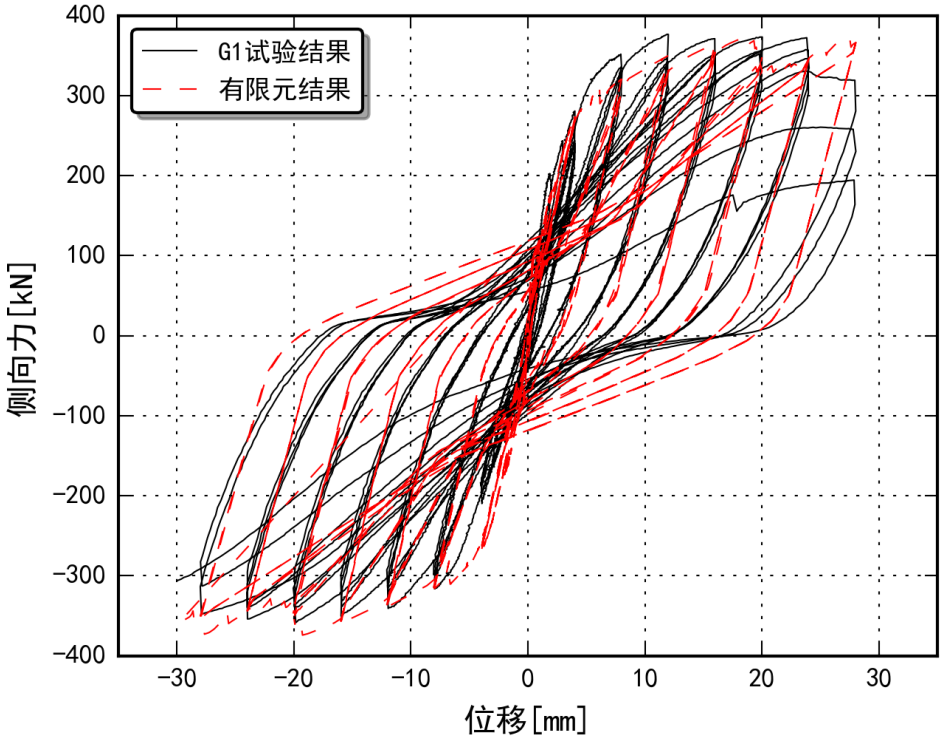
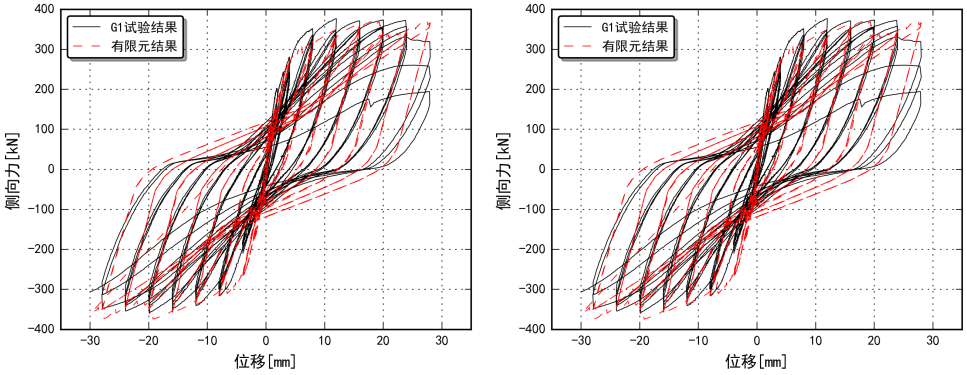



图 6-1 G1 数据对比




(a) G1 试件 (b) G1 试件

图 6-2 太合金多炭钢铁产品柱扭曲局部受力分析示意图

## 6.2 公式测试

公式的排版和引用与正常 LaTeX 公式格式一致。例如：

$$e^{i\pi}+1=0 \tag{6-1}$$

欧拉公式 (式**??**)，它是数学里最令人着迷的一个公式，它将数学里最重要的几个数学联系到了一起——两个超数：自然对数的底 $e$，圆周率 $\pi$；两个单位：虚数单位 $i$ 和自然数的单位 1，以及数学里常见的 0。数学家们评价它是“上帝创造的公式”，我们只能看它而不能理解它。

## 6.3 参考文献测试

我使用的参考文献是符合 GB7714-2015 规范的中文论文参考文献格式。该规范作为国家标准化管理委员会正式公布的标准，其权威性和通用性远远超过了其它格式规范，是国内最正式的参考文献格式规范，并且是大多数高校毕业论文和杂志的指定参考文献格式。

下面分类测试各种类型参考文献：

- 中文规范[C1]。
- 英文规范[ACI318]
- 中文学位论文[liguiqian]
- 英文学位论文[bentz2000]
- 期刊[FMK]
- 图书类[B1]
- 其他[R1]
- 多文献引用，压缩格式[C1,ACI318,liguiqian,R1]

1. one
2. two
3. ...

# 7 结论

论文的结论是最终的、总体的结论，不是正文中各段的小结的简单重复。结论应该准确、完整、明确、精练。如果不可能导出应有的结论，也可以没有结论而进行必要的讨论。可以在结论或讨论中提出建议、研究设想、仪器设备改进意见以及尚待解决的问题等。

# 致　　谢

放置在摘要页前，对象包括：1）国家科学基金，资助研究工作的奖学金基金，合同单位，资助或支持的企业、组织或个人。2）协助完成研究工作和提供便利条件的组织或个人。3）在研究工作中提出建议和提供帮助的人。4）给予转载和引用权的资料、图片、文献、研究思想和设想的所有者。5）其他应感谢的组织和个人。

# 附　录

## 附录 A　外文文献原文

MIPRO 2012, May 21-25,2012, Opatija, Croatia

A brief introduction to OpenCV

Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo, Mario Cifrek
Faculty of electrical engineering and computing, University of Zagreb, Zagreb, Croatia
ivan.culjak@fer.hr, david.abram@fer.hr, tomislav.pribanic@fer.hr,hrvoje.dzapo@fer.hr,mario.cifrek@fer.hr

**Abstract** - The purpose of this paper is to introduce and quickly make a reader familiar with OpenCV (Open Source Computer Vision) basics without having to go through the lengthy reference manuals and books. OpenCV is an open source library for image and video analysis, originally introduced more than decade ago by Intel. Since then, a number of programmers have contributed to the most recent library developments. The latest major change took place in 2009 (OpenCV 2) which includes main changes to the C++ interface. Nowadays the library has >2500 optimized algorithms. It is extensively used around the world, having >2.5M downloads and >40K people in the user group. Regardless of whether one is a novice C++ programmer or a professional software developer, unaware of OpenCV, the main library content should be interesting for the graduate students and researchers in image processing and computer vision areas. To master every library element it is necessary to consult many books available on the topic of OpenCV. However, reading such more comprehensive material should be easier after comprehending some basics about OpenCV from this paper.

**I. INTRODUCTION**

Computer Vision is the science of programming a computer to process and ultimately understand images and video, or simply saying making a computer see [1]. Solving even small parts of certain Computer Vision challenges, creates exciting new possibilities in technology, engineering and even entertainment. In order to advance vision research and disseminate vision knowledge, it is highly critical to have a library of programming functions with the optimized and portable code, and hopefully available for free. This was an original goal of Intel team back in 1999 when OpenCV (Open Source Computer Vision Library) was officially launched. Since then, a number of programmers have contributed to the most recent library developments. The latest major change took place in 2009 (OpenCV 2) which includes main changes to the C++ interface. The newest library release can be found on the OpenCV official website [2]. Nowadays the library has >2500 optimized algorithms. It is extensively used around the world, having >2.5M downloads and >40K people in the user group. OpenCV can be used in academic and commercial applications as well, under a BSD license [3]. To master every OpenCV library element it is necessary to consult many books available on the topic of OpenCV. Nevertheless, reading such more comprehensive material should be easier after comprehending a basic idea about OpenCV from this paper. In fact to make it even more convenient, the text presented here intentionally closely follows one of the most recent OpenCV sources [4].

**II. BASIC LIBRARY STRUCTURE AND FEATURES**

OpenCV library (since version 2.2) is divided into several modules, where each module can be understood, in general, as being dedicated to one group of computer vision problems. All the classes and functions are defined within the name space cv. Therefore to access them we can either precede the main function definition by the declaration using namespace cv; or prefix OpenCV class and function names by namespace specification cv::. The main object is of class Mat. As implicated by the class name it is essentially a matrix holding pixel values of some image and, in addition, a number of attributes about an image. In the simplest case an image can be created as cv:: Mat image;, creating an image of size 0 by 0. Perhaps the most important member variable of image object is data where image.data member is actually a pointer to the allocated memory block that contains the image data (in this trivial case it would be image.data=0). Alternatively during a creation of Mat object we can explicitly specify an initial size and the type of each matrix element. This type specifies, for example, signed 1-byte pixel image values (CV_8U), or three channels for a color image (CV_8UC3), or even 32-bit/64bit floating point numbers (CV_32F).

Once an object of class mat is defined, a nice feature about it (not present in the early versions of OpenCV) is that a memory allocation/deallocation takes place automatically. For instance, a memory automatically allocated during the image read out into some object, will be also automatically released once the corresponding object goes out of scope.

Another important thing is that Mat class implements the reference counting and shallow copy. Hence, when an image is assigned to another one, the image data itself is not copied and both images point to the same memory block (this also applies to images passed by/returned by value). However, since reference count is supported, a memory allocated for image (pixel) data itself will be released only when all of the references to the image are destructed.

In the versions prior to OpenCV 2, C like functions and structures were used (still can be though) and the main structure was IplImage. Although there is a convenient way to convert IplImage structure into cv::Mat object, it is highly recommended to avoid this deprecated data structure.

# 附录 B 外文文献翻译

MIPRO 2012，2012 年 5 月 21 日至 25 日，克罗地亚奥帕蒂亚

OpenCV 简介

伊万·库尔贾克（Ivan Culjak），大卫·艾布拉姆（David Abram），托米斯拉夫·普里巴尼奇（Tomislav Pribanic），赫尔沃耶·德扎波（Hrvoje Dzapo），马里奥·奇夫雷克（Mario Cifrek）
克罗地亚·萨格勒布，萨格勒布大学·电气工程与计算学院
ivan.culjak@fer.hr，david.abram@fer.hr，tomislav.pribanic@fer.hr，rvoje.dzapo@fer.hr，
mario.cifrek@fer.hr

**摘要**——本文的目的是介绍并使读者快速熟悉 OpenCV（开源计算机视觉库）的基础知识，而无需翻阅冗长的用户手册和书籍。OpenCV 是一个开源的、用于图像和视频分析的库，最初由英特尔

在十几年前提出。从那以后，大量程序员对最新的这版库做出了贡献。最新的重大变化发生在 2009 年（OpenCV 2 版本），这一版本包含了对 C ++ 接口的主要修改。如今这个库有超过 2500 个优化算法。它被广泛用于全球各地，下载量超过 250 万，用户组超过 4 万人。对于研究图像处理和计算机视觉领域的研究生和研究人员而言，不管是 C ++ 的新手程序员还是专业软件开发人员，很少不知道 OpenCV，这个领域的主要的库内容。要想掌握每一个库元素，有必要查阅许多 OpenCV 领域的书籍。但是，理解本文中有关 OpenCV 的一些基础知识后，再去读更全面的材料应该会更容易。

**一、简介**

计算机视觉是一门编写计算机程序来处理并最终理解图像和视频，或者简单说来，让计算机可以“看到”的科学。哪怕解决计算机视觉的很小一部分挑战，就可以在技术，工程学甚至娱乐领域创造令人兴奋的新机遇。为了推进视觉研究并传播视觉知识，建立一个优化的、可移植的，而且最好是可以免费使用的程序函数的代码库是非常必要的。时间倒流回 1999 年，OpenCV（开源计算机视觉库）正式发布的时候，这就是英特尔团队最初的目标。从那时起，许多程序员为最新一版的库的开发做出了贡献。最新的重大变化发生在 2009 年（OpenCV 2）发布的时候，它包含了对 C++ 接口的主要更改。最新的库的发行版可以在 OpenCV 的官方网站上找到。如今，该库有 2500 多个优化算法。它在世界范围内被广泛使用，下载量超过 250 万，用户数量超过 4 万。OpenCV 可用于学术领域，也可以在 BSD 许可下用于商业领域。要想掌握每一个库元素，有必要查阅许多 OpenCV 领域的书籍。但是，理解本文中有关 OpenCV 的一些基础知识后，再去读更全面的材料应该会更容易。实际上，为了使它更加易读，这里的用语和措辞遵循最新的 OpenCV 源码版本之一。

**二、库的基本结构和特性**

OpenCV 库（自版本 2.2 起）分为几个模块，通常情况下，每个模块都可以理解为专用于解决一组计算机视觉的问题。所有的类和函数都已定义在命名空间 cv 中。因此，为了访问它们，我们可以在主函数定义之前加上 using namespace cv；或在 OpenCV 类和函数名称前加前缀，通过命名空间 cv :: 声明。主要的对象是 Mat 类。就像类名所提示的那样，它本质上是一个矩阵，用于存放一些图像的像素值，此外，还有一些图像的基本属性。在最简单的情况下，可以将图像创建为 cv :: Mat image;，这就创建了一个大小为 0 x 0 的图像。image 对象最重要的成员变量是 data，image.data 成员实际上是指向分配的包含图像数据的内存块的指针（在这种简单的情况将是 image.data= 0）。另外，在创建 Mat 对象的过程中，我们可以显式指定每个矩阵元素的初始大小和类型。例如可以指定，带符号的 1 字节像素图像值（CV_8U），或彩色图像的三个通道（CV_8UC3），甚至 32 位/ 64 位浮点数（CV_32F）。定义了 mat 类的对象后，它的一个很好的特性是（在早期版本的 OpenCV 中不存在）是自动进行内存分配/释放。例如，在图像读取期间内存自动分配到某个对象中，同时，一旦对应对象超出生命周期，也会自动释放。

另一个重要的事情是 Mat 类实现了参考计数和浅表复制。因此，当图像内容被分配给另一个图像，图像数据本身并未复制，两个图像都指向相同的内存块（这也适用于传递/返回的图像值）。但是，由于支持引用计数，因此分配给图像（像素）数据本身的内存仅当所有对图像的引用均被销毁时才释放。在 OpenCV 2 之前的版本中，类似 C 的函数和结构被使用（虽然仍然可以），主要结构是 IplImage。虽然有一个将 IplImage 结构转换为 cv :: Mat 对象的便捷方法，但是强烈建议避免使用不推荐的数据结构。