本科毕业论文

论文题目（中文） 基于深度学习的乳腺癌检测

论文题目（英文） Breast cancer detection based on deep learning

学生姓名 姚忱

指导教师 张红娟

学　　院 信息科学与工程学院

专　　业 电子信息科学与技术

年　　级 2019 级

兰州大学教务处

诚信责任书

本人郑重声明：本人所呈交的毕业论文（设计），是在导师的指导下独立进行研究所取得的成果。毕业论文（设计）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人、集体已经发表或未发表的论文。

本声明的法律责任由本人承担。

论文作者签名：_____ 日 期：_____

关于毕业论文（设计）使用授权的声明

本人在导师指导下所完成的论文及相关的职务作品，知识产权归属兰州大学。本人完全了解兰州大学有关保存、使用毕业论文（设计）的规定，同意学校保存或向国家有关部门或机构送交论文的纸质版和电子版，允许论文被查阅和借阅；本人授权兰州大学可以将本毕业论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用任何复制手段保存和汇编本毕业论文（设计）。本人离校后发表、使用毕业论文（设计）或与该毕业论文（设计）直接相关的学术论文或成果时，第一署名单位仍然为兰州大学。

本毕业论文（设计）研究内容：

☑可以公开

□ 不宜公开，已在学位办公室办理保密申请，解密后适用本授权书。

（请在以上选项内选择其中一项打“√”）

论文作者签名：_____ 导师签名：_____

日 期：_____ 日 期：_____

基于深度学习的乳腺癌检测

中文摘要

本项目使用了深度学习技术来实现乳腺癌的早期诊断。项目是在 INbreast，BraeakHis 等乳腺癌图像数据集上训练 CNN 模型来对乳腺癌病例进行分类。比较了不同的数据集，不同的 CNN 模型及不同的分类方式的表现，并在验证集上使用准确度等指标评估模型的性能。最终，项目提供了一个使用 Keras 实现的 CNN 模型，该模型可以准确地分类乳腺癌图像

关键词：乳腺癌，深度学习，卷积神经网络

Breast cancer detection based on deep learning

Abstract

This project uses deep learning technology to achieve early diagnosis of breast cancer. The project is to train a CNN model on INbreast, BraeakHis and other breast cancer image datasets to classify breast cancer cases. The performance of different datasets, different CNN models and different classification methods is compared, and the performance of the model is evaluated using indicators such as accuracy on the verification set. Ultimately, the project provides a CNN model implemented using Keras that can accurately classify breast cancer images.

Keywords: Breast Cancer;Deep Learing;Convolutional Neural Network

目　录

图　目　录

表　目　录

第一章 绪 论

1.1 问题背景及研究意义

乳腺癌是全球女性最为常见的恶性肿瘤之一。据国家癌症中心最新统计，全国每年新发乳腺癌病例数达 27.24 万，死亡人数超过 7 万。其发病率呈现明显的上升趋势，高居女性恶性肿瘤发病率首位，已经成为女性健康的巨大威胁。自上世纪八十年代以来，乳腺癌在中国的发病率持续快速上升，并呈现年轻化趋势，这对女性健康造成了严重的威胁。早期发现和诊断乳腺癌是治愈该病的关键，因此乳腺癌检测技术备受瞩目。

然而，传统的乳腺癌检测技术，如组织病理学检查，乳房 X 线摄影和超声波检查等，存在着诸多限制，如良恶性病变的区分困难和漏诊率高等问题。随着深度学习技术的不断发展，越来越多的学者开始研究基于深度学习的乳腺癌检测技术，以期改善传统技术所存在的局限性。

第二章 基本原理与方法介绍

2.1 研究方法

乳腺癌的诊断一直依赖于传统的组织病理学检查。然而，病理学家对组织图像进行人工分析是一项耗时且繁琐的过程，其诊断结果存在较大的主观性和误差风险。

因此，本研究旨在利用深度卷积神经网络（CNN）模型对乳腺癌图像进行分类，以提高乳腺癌诊断的准确性和效率。我们使用不同的乳腺癌图像数据集来训练 CNN 模型，并对分类结果进行比较分析。

2.2 卷积神经网络简介

卷积神经网络（Convolutional Neural Network，CNN）是一种广泛应用于计算机视觉领域的人工神经网络模型。CNN 的基本思想是将输入图像作为网络的输入，并通过一系列卷积层、池化层、全连接层、非线性激活函数等操作，提取出图像的特征并进行分类或识别等任务。CNN 的主要优点是它可以自动学习输入数据的特征，从而无需手动设计特征提取器，因此被广泛应用于图像识别、目标检测、人脸识别、自然语言处理等各个领域。

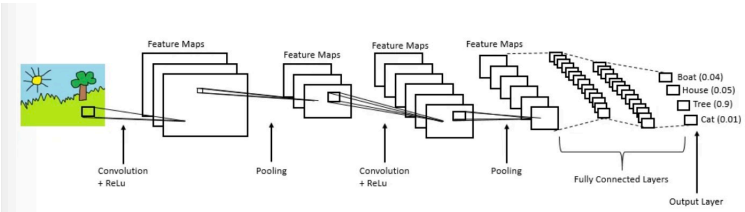

图 2.1 CNN 基本结构

2.3 卷积神经网络基本结构

2.3.1 卷积层

卷积神经网络（Convolutional Neural Network，CNN）中的卷积层是其核心部分之一。卷积层通过卷积运算对输入进行特征提取，并生成输出特征图。每个卷积层通常由多个过滤器（也称为卷积核）组成，每个过滤器都是一个小的可学习的权重矩阵，用于在输入上进行卷积运算，从而提取出局部特征。卷积运算通常会对输入的每个像素及其周围的像素进行权重相乘和加和的操作，从而生成一个新的特征图，该特征图对输入的局部区域进行了下采样和特征提取，且保留了局部位置信息。

在 CNN 中，每个卷积层的过滤器数量和大小通常是预先设定的，而过滤器的权重则通

过反向传播算法进行训练。在训练过程中，CNN 会通过反向传播来调整过滤器的权重，以最小化输出特征图与标签之间的差距。通过反复的训练，CNN 可以逐渐优化卷积层的过滤器，使其能够提取出更加抽象和高级的特征，从而提高模型的性能和泛化能力。

卷积层的计算公式可以表示为：

$$y_{i,j} = \sum_{m=0}^{k-1}\sum_{n=0}^{k-1} x_{i+m,j+n} \cdot w_{m,n} + b$$

其中，x 表示输入特征图，y 表示输出特征图，w 表示卷积核权重矩阵，b 表示偏置项，k 表示卷积核的大小。

具体来说，对于输出特征图中的每一个像素点 (i,j)，我们会将该像素点周围的 $k \times k$ 个像素点与卷积核中的权重矩阵进行卷积操作，即对于卷积核中的每一个权重 $w_{m,n}$，将输入特征图中位置为 $(i+m, j+n)$ 的像素点乘以该权重，然后将所有乘积加起来，并加上偏置项 b，即为该像素点在输出特征图中的值 $y_{i,j}$。这个过程可以看做是对输入特征图进行局部的线性变换和下采样，从而提取出输入特征图的局部特征。实现如下图所示：

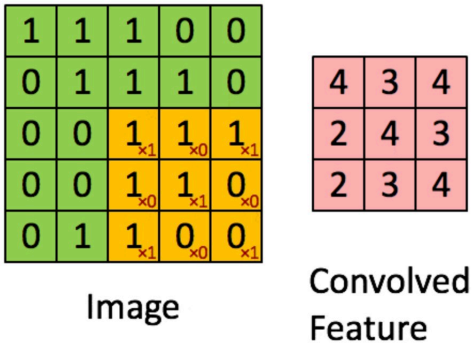

图 2.2　卷积层实现

2.3.2　池化层

池化层是卷积神经网络中一种常用的层次结构，

池化操作通常分为最大池化和平均池化两种，其中最大池化是取输入特征图中每个子区域的最大值，而平均池化是取每个子区域的平均值。池化操作的参数通常由池化窗口大小和步幅大小两个参数决定。池化窗口大小指的是取样区域的大小，而步幅大小指的是池化窗口在特征图上移动的距离。

池化层的作用是降低特征图的维度，减少计算量和模型参数数量，从而防止过拟合。池化层可以对每个卷积核提取的特征图进行下采样操作，从而在保留特征信息的同时，将特征图的尺寸降低一定程度上，增强模型的泛化能力。

最大池化的计算公式为：

$$y_{i,j} = \max_{m,n}(x_{(i-1)s+m,(j-1)s+n})$$

其中，x 是输入特征图，y 是输出特征图，s 是池化窗口的步幅大小，m 和 n 是池化窗口内的像素坐标。

平均池化的计算公式为：

$$y_{i,j} = \frac{1}{k^2}\sum_{m=1}^{k}\sum_{n=1}^{k}x_{(i-1)s+m,(j-1)s+n}$$

其中，k 是池化窗口的大小。

以最大池化为例，实现过程如下图：

图 2.3　池化层实现

2.3.3　全连接层

在卷积神经网络中，全连接层是一种用于对特征进行分类或者回归的层次结构。全连接层接收前面所有层次的输出，将其展开为一个一维向量，并将其与一组权重进行乘积和偏置相加，最终输出一个新的向量，用于进行分类或回归任务。

在卷积神经网络的末尾，通常会添加一个或多个全连接层，用于将卷积层和池化层提取的特征进行整合，并输出一个最终的分类或回归结果。全连接层的参数数量通常比较大，需要进行大量的参数学习，因此容易导致过拟合问题，需要进行一定的正则化操作来降低模型的复杂度。

全连接层的计算方式比较简单，它的输出可以通过以下公式来计算：

$$y = Wx + b$$

其中，x 是输入向量，W 是权重矩阵，b 是偏置向量，y 是输出向量。权重矩阵 W 的大小为 $N \times M$，其中 N 是输出向量的大小，M 是输入向量的大小，偏置向量 b 的大小为 N。

全连接层的输出可以经过激活函数进行非线性变换，从而增加模型的表达能力。常用的激活函数包括 sigmoid 函数、ReLU 函数、tanh 函数等。全连接层的输出也可以经过 dropout 等正则化操作来减少过拟合问题。

全连接层实现如下图所示：

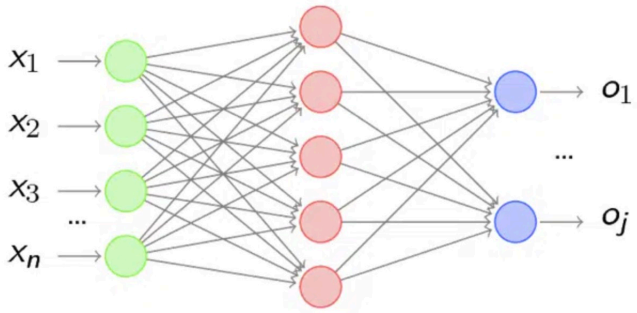

图 2.4 卷积层实现

2.4 研究所用到的神经网络模型

2.4.1 VGG16

VGG16 是牛津大学的 K.Simonyan 和 A.Zisserman 在论文“Very Deep Convolutional Networks for Large-Scale Image Recognition”中提出的卷积神经网络模型。

其网络结构具有一定的规律性。该网络接受 224×224 的 RGB 图像作为输入，经过一系列的卷积层和最大池化层，输出一个 1000 维的向量，表示图像的类别概率。

VGG16 网络包含 13 个卷积层和 3 个全连接层，其中卷积层都采用 3×3 的卷积核进行特征提取，并使用 ReLU 激活函数进行非线性变换。卷积层之后的最大池化层通过对特征图进行下采样，提取最显著的特征。

在网络的末尾，VGG16 采用三个全连接层进行分类。前两个全连接层各自有 4096 个神经元，最后一个全连接层有 1000 个神经元，每个神经元对应于一个类别。最后一层是 softmax 层，将输出向量转化为类别概率。

VGG16 结构如下：

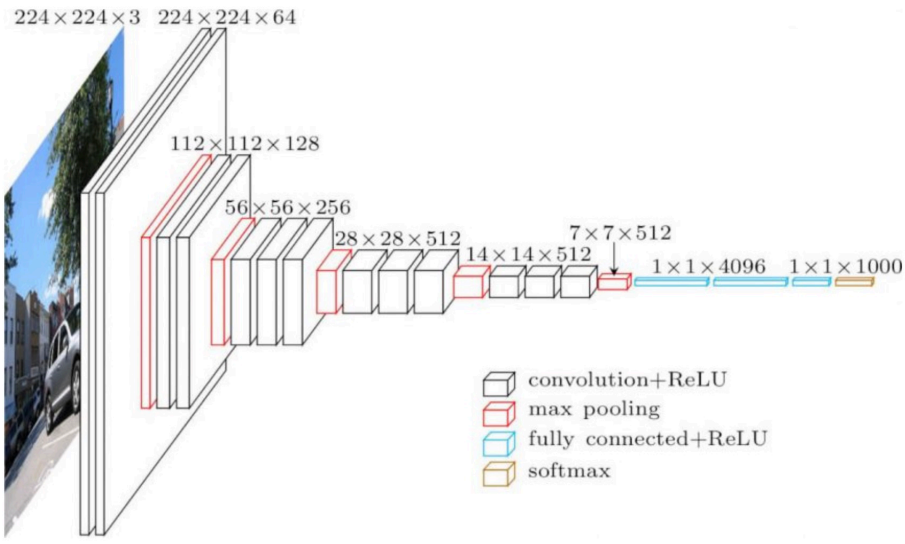

图 2.5 VGG16

2.4.2 ResNet50

ResNet50 是一个经典的深度卷积神经网络，由何凯明等人于 2015 年提出，是 ResNet 系列中较为流行的一个网络结构。ResNet50 共包含 50 个卷积层，其中包括 16 个 Bottleneck 块和一个全局平均池化层，最后连接一个具有 1000 个输出的全连接层，用于 ImageNet 分类任务。

ResNet50 引入了残差学习的思想，通过在网络中加入残差块来缓解梯度消失问题。每个残差块包含两个 3x3 卷积层和一个跨层连接，其中跨层连接绕过了一个或多个卷积层，将输入直接添加到输出上。这种跨层连接方式使得模型可以更好地捕捉到输入特征的变化，同时避免了梯度消失的问题，使得网络训练更加容易。

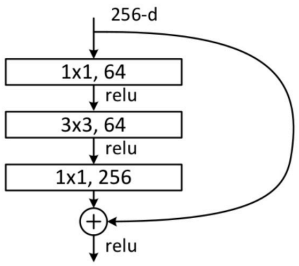

图 2.6 VGG16

ResNet50 在许多计算机视觉任务中都表现出色，不仅在 ImageNet 分类任务中取得了优异的成绩，同时在物体检测、语义分割等任务中也广泛应用。同时，ResNet50 也为后来的深度神经网络提供了很多启示，如更深层次的 ResNet-101、ResNet-152 等。

2.4.3 DenseNet201

2.5 算法流程

算法主要流程如下：1. 导入所需的 Python 库。

2. 使用 Dataset$_{l}oaderjpg$

3. 从训练和验证数据中的良性和恶性皮肤癌病例中创建标签，合并数据并将其打乱。

4. 将标签转换为分类矩阵。

5. 使用图像数据生成器 ImageDataGenerator 从训练集中生成批处理数据。

6. 创建一个具有 DenseNet201 和 GlobalAveragePooling2D 层的模型，通过使用 Adam 优化器和分类交叉熵损失函数来编译该模型。

7. 使用生成器对模型进行训练，设置回调函数以在验证集上监控模型的性能并定期保存最佳模型。

8. 在验证集上进行预测并评估模型性能。

9. 使用测试集和 Test Time Augmentation (TTA) 进行预测并生成最终的分类结果。

第三章 实验和分析

3.1 数据集

3.1.1 BreakHis 数据集

乳腺癌组织病理学图像数据集 BreaKHis，收集自 82 位患者的乳腺肿瘤组织样本，共包含 9,109 幅显微图像，分别使用不同放大倍数（40X、100X、200X 和 400X）获得。该数据集目前已包含 2,480 个良性样本和 5,429 个恶性样本，图像大小为 700×460 像素，以 3 通道 RGB 格式存储，每个通道有 8 位深度，采用 PNG 格式保存。

总体而言，BreaKHis 数据集可分为两个主要类别：良性肿瘤和恶性肿瘤。组织学良性指的是不符合任何恶性标准的病变，例如，细胞形态正常、无明显的有丝分裂、基底膜未破裂和未发生转移等。通常，良性肿瘤的生长缓慢、范围有限，相对较为“温和”。恶性肿瘤是癌症的同义词，该病变可侵犯并破坏周围的结构（局部浸润），并向远处扩散（转移），最终导致患者死亡。

BreakHis 数据集结构如下：

表 3.1 BreakHis 数据集结构

Magnification	Benign	Malignant	Total
40X	652	1,370	1,995
100X	644	1,437	2,081
200X	623	1,390	2,013
400X	588	1,232	1,820
Total of Images	2,480	5,429	7,909

而根据肿瘤细胞在显微镜下的外观，又可以将良性和恶性乳腺肿瘤分为不同的类型。各种类型/亚型的乳腺肿瘤可能具有不同的诊断和治疗意义。该数据集目前包含四种不同组织学类型的良性乳腺肿瘤：腺病 (A)、纤维腺瘤 (F)、叶状肿瘤 (PT) 和管状腺酮 (TA)；和四种恶性肿瘤（乳腺癌）：癌（DC）、小叶癌（LC）、粘液癌（MC）和乳头状癌（PC）。

据此在本文在实现乳腺癌图像分类时采用了不同的分类标准，并对结果进行了比较。一是最基础的二分类，即将所有图像仅分成良性恶性两类。

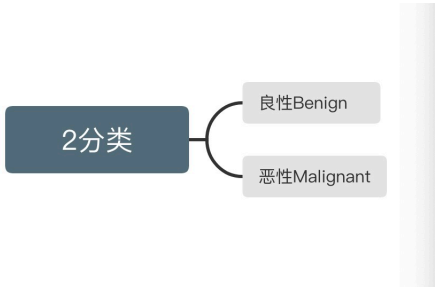

图 3.1　2 分类

二是直接进行 8 分类，即直接将整个数据集分成 A，F，PT，TA，DC，LC，MC，PC8 类。

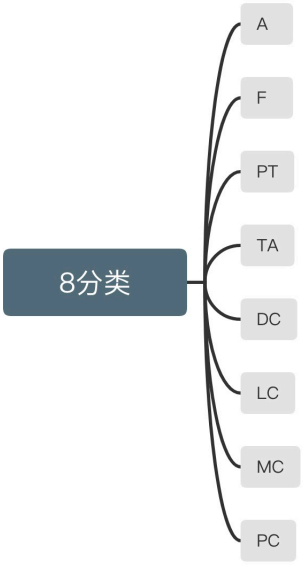

图 3.2　8 分类

三是分别对良性和恶性两个数据集进行四分类。

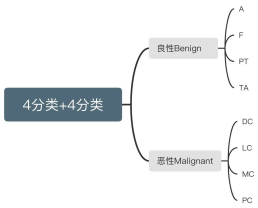

图 3.3　4 分类 +4 分类

3.1.2 INbreast 数据集

INbreast 是一个用于数字乳腺 X 射线图像识别和分割的公共数据集，其中包含了 410 张数字乳腺 X 射线图像和对应的乳腺组织分割掩模。这个数据集可以用于乳腺癌筛查和诊断等医学研究领域。该数据集中的图像是由印度国立癌症研究所的专业医生进行标注和诊断的。

INbreast 数据集中的每张乳腺 X 射线图像都包含了医生的诊断结果，包括该乳腺病变是否为恶性（乳腺癌）等信息。该数据集中的图像被标记为“良性”或“恶性”，并且每个标注都由两名专业医生进行独立审核和诊断，以确保标注的准确性和可靠性。同时，数据集中也包含了一些异常和其他诊断结果的标注，例如乳腺增生和钙化等。这些诊断结果和标注信息可以帮助研究人员进行乳腺癌筛查和诊断算法的开发和评估。

INbreast 数据集中的每张图像的标注信息可以通过查看相应的 XML 文件来获取。每个 XML 文件都与该图像的文件名相对应，并且包含了该图像的详细标注信息，例如病变的位置、大小、形状、密度等等。同时，每个 XML 文件中也包含了该图像的诊断结果，即该图像是否为恶性乳腺癌等信息。可以使用 XML 文件解析器等工具来读取和解析这些 XML 文件，以获取乳腺 X 射线图像的标注信息和诊断结果。此外，INbreast 数据集的官方网站上也提供了相关的文档和代码示例，以帮助研究人员更好地使用和理解该数据集。

3.2 实验结果分析

3.2.1

首先在选用 BreakHis 数据集进行二分类时，整个数据集进行如下分割：

```
dataset train
  benign
   b1.jpg
   b2.jpg
   //
  malignant
   m1.jpg
   m2.jpg
   //  validation
   benign
    b1.jpg
    b2.jpg
    //
   malignant
    m1.jpg
    m2.jpg
    //...
```

图 3.4 4 分类 +4 分类

由于数据集中良性与恶性的样本数量不匹配，以以下两种方法进行实验。一种是使用原始的数据集，即将测试集和训练集中包含整个数据集中的所有样本。二是对整个数据集进行抽取，使得良性和恶性样本的数量相匹配。具体抽取规则为训练集良性与恶性每个类别各有 1000 张图像，而验证文件夹每个类别各有 250 张图像。

```
benign_train = np.array(Dataset_loader('drive/MyDrive/data/train/benign',224))
malign_train = np.array(Dataset_loader('drive/MyDrive/data/train/malignant',224))
benign_test = np.array(Dataset_loader('drive/MyDrive/data/validation/benign',224))
malign_test = np.array(Dataset_loader('drive/MyDrive/data/validation/malignant',224))
```

```
100%|██████████| 1000/1000 [00:23<00:00, 42.28it/s]
100%|██████████| 1000/1000 [00:22<00:00, 43.83it/s]
100%|██████████| 250/250 [00:09<00:00, 25.53it/s]
100%|██████████| 250/250 [00:07<00:00, 32.18it/s]
```

图 3.5 抽取后的数据集

```
benign_train = np.array(Dataset_loader('drive/MyDrive/data_origin/train/benign',224))
malign_train = np.array(Dataset_loader('drive/MyDrive/data_origin/train/malignant',224))
benign_test = np.array(Dataset_loader('drive/MyDrive/data_origin/validation/benign',224))
malign_test = np.array(Dataset_loader('drive/MyDrive/data_origin/validation/malignant',224))
```

```
100%|██████████| 2392/2392 [02:27<00:00, 16.26it/s]
100%|██████████| 5328/5328 [04:11<00:00, 21.19it/s]
100%|██████████| 100/100 [00:28<00:00,  3.52it/s]
100%|██████████| 100/100 [00:31<00:00,  3.13it/s]
```

图 3.6 原数据集

通过实验，经过抽取后的

1. 正文 AHEMoS$\alpha\beta$
2. 公式 $AHEMoS\alpha\beta$
3. mathbf $\mathbf{AHEMoS}\alpha\beta$
4. boldsymbol $\boldsymbol{AHEMoS\alpha\beta}$

这个加粗、斜体、英文字体（含正文和公式内字体），有不同的处理方式，在.cls 模板文件文件搜索 bm 查看详细说明

3.2.2 左边大括号

$$\begin{cases} \vec{e}_1 &= \frac{3a}{2}\vec{i} + \frac{\sqrt{3}a}{2}\vec{j} \\ \vec{e}_2 &= \frac{3a}{2}\vec{i} - \frac{\sqrt{3}a}{2}\vec{j} \end{cases} \tag{3.1}$$

注意后面有个方程的编号，如果想取消，把上下的两个 *equation* 改成 *equation*∗

$$\begin{cases} \vec{e}_1 &= \frac{3a}{2}\vec{i} + \frac{\sqrt{3}a}{2}\vec{j} \\ \vec{e}_2 &= \frac{3a}{2}\vec{i} - \frac{\sqrt{3}a}{2}\vec{j} \end{cases}$$

3.2.3 复杂公式

不会输出的符号，请百度，啥都有

$$H\hat{Q}R = \frac{\epsilon}{2}\hat{\sigma}_z - \frac{\Delta}{2}\hat{\sigma}_x + \sum_k \omega_k \hat{b}_k^\dagger \hat{b}_k + \sum_k \frac{g_k}{2}\hat{\sigma}_z(\hat{b}_k + \hat{b}_k^\dagger) \tag{3.2}$$

3.2.4 等号对齐站

主要是用这个 aligned 放在了方程的环境里，等号前面 & 控制对齐，每一行后面双斜杠换行

$$\begin{aligned}\vec{CH} &= m\cdot\vec{e}_1 + n\cdot\vec{e}_2 \\ &= \frac{3(m+n)a}{2}\vec{i} + \frac{\sqrt{3}(m-n)a}{2}\vec{j}\end{aligned} \tag{3.3}$$

3.2.5 矩阵乘法

其实就是几个 array 组合

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.4}$$

3.2.6 附页代码

可以在 LZUThesis.clc 里面修改代码格式

java 代码

```
System.out.print("兰朵儿")
// 试一下中文注释
```

tex 代码

```
width=0.3\textwidth
% 注释
```

python 代码

```
print("兰朵儿")
# 注释
```

matlab 代码有专门的库，但是没必要高亮太多，而且中文适配有问题，直接按照下面这个就可以

```
display("兰朵儿")
% 注释
```

伪代码

算法 1 PMHSS 算法

给定一个初值 $x^{(0)} \in C^n$ 和常数 $\alpha > 0$
for $k = 1, 2, \ldots$ 直到序列 $\{x^{(k)}\}_{k=0}^{\infty}$ 收敛 **do**
    解方程: $(\alpha V + W)x^{(k+\frac{1}{2})} = (\alpha V - iT)x^{(k)} + b$
    解方程: $(\alpha V + T)x^{(k+1)} = (\alpha V + iW)x^{(k+\frac{1}{2})} - ib$
end for

3.2.7 参考文献

这个，百度学术、谷歌学术等网站都可以导出 bibtex 格式的参考文献（知网不行，网上有个人写了个转换器，但是 windows 用不了，就不放了，尽量用谷歌学术把那个文献找出来吧），直接放在 bib/database.bib 文件里、知网需要用其他东西转换，但是我建议用 mendeley 这个软件管理文献，然后可以导出 bibtex 格式的，甚至可以直接复制引用，很方便 [**?**, **?**, **?**]。

有些人希望多个参考文献同时引用时用 [1-3] 而不是 [1,2,3],所以我加了个包 cite。(2020-5-18)

具体怎么用可以百度，我这里告诉你什么可以用，但是具体的，建议百度，更靠谱一些。

有参考文献时，编译要经过 4 步，直接 XeLaTeX --> BibTeX --> XeLaTeX --> XeLaTeX，不然很多问题，vscode 配置以后很方便，以下内容放在设置中，重新打开 vscode 即可

修改后可以参考文献自动生成中文等字符 [**?**] [**?**] [**?**]，引用网络资源时链接格式规范化 [**?**, **?**]。

测试右上角 [?]

中英文参考文献说明

感谢的代码贡献 潘麒

进一步说明，对于中文参考文献，建议添加条目 language= 中文 这一行，否则多个作者，不是“等.”[**?**] 而是“et al.”[**?**]

```
1
2     @Article{partl2021,
3     author = {Partl, Hubert and Hyna, Irene  and 兰朵儿 and
          Schlegl, Elisabeth},
4     title  = {一个中文等测试},
5     year   = {2021},
6     language = {中文},
7     journal = {测试期刊},
8     volume={3},
```

```
9       number={6},
10      pages={10--20},
11    }
```

3.2.8 引用图、表、公式、章节

为什么要引用？不直接写数字？因为图表顺序变化时，引用的地方会自动变化。每次更加新引用，请四步走编译

引用的地方加 label，自己写个名字，可以是中文，然后引用的地方如下：

如图**??**所示

如公式(**??**)所示，会自动带括号

如表**??**所示

在**??**中已经提及

附　　录

这里是附录页，附上你的程序或必要的相关知识

编译方式: XeLaTeX -->BibTeX --> XeLaTeX-->XeLaTeX

致　　谢

这里是致谢页，你可以在这里致谢你的舍友，老师，朋友，或者我

（我是谁？兰朵儿开发者：余航，致谢我，查重时一定会重复的，哈哈，开个玩笑，本科生论文不在查重范围，而且“毕业论文(设计)检测内容主要为毕业论文(设计)的主体部分”）。

毕业论文（设计）成绩表

导师评语

导师评价你人很好

建议成绩_____ **指导教师（签字）**_____

答辩委员会意见

优秀

答辩委员会负责人（签字）_____

成绩 **100** **学院（盖章）**_____

年 月 日