

# 模型创新及相关说明

## 模型创新及相关说明

- 0. 项目运行说明
- 1. BIRADS分类
  - 1.1. 模型原理
    - 1.1.1. Encoder 编码器
    - 1.1.2. 解码器
  - 1.2. 数据预处理
  - 1.3. 实现细节
  - 1.4. 创新点
- 2. 特征识别
  - 2.1 模型原理
  - 2.2 数据预处理
  - 2.3 实现细节
  - 2.4 创新点

## 0. 项目运行说明

- 环境参考：

框架名称	框架版本	Cuda 名称	Python 名称
PyTorch >	2.4.0 >	12.1.1 >	3.11
TensorFlow >	2.2.1 >		3.11
PaddlePaddle >	2.0.0 >		
OneFlow >	1.13.1 >		
OpenCV >	1.13.0 >		
MXNet >	1.12.1 >		
KataGo >	1.12.0 >		

请选择框架名称/框架版本/Cuda版本/Python版本 ^

- 服务器配置参考：
  - 显卡类型：3090-24G
  - GPU数量：1
  - CPU型号：Intel(R) Xeon(R) CPU E5-2686 v4
- 框架参考（同requirements.txt）：

```
1 matplotlib==3.8.3
2 numpy==1.26.4
3 Pillow==10.3.0
4 scikit-learn==1.5.0
5 torch==2.0.0
6 torchvision==0.15.1
7 torchaudio==2.0.0
8 tqdm==4.66.2
9 opencv-python
10 pandas
```

# 1. BIRADS分类

---

- 对于超声乳腺图像的BIRADS分类问题，我们使用传统Vision Transformer模型进行图像分类

## 1.1. 模型原理

- 其原理图如下：

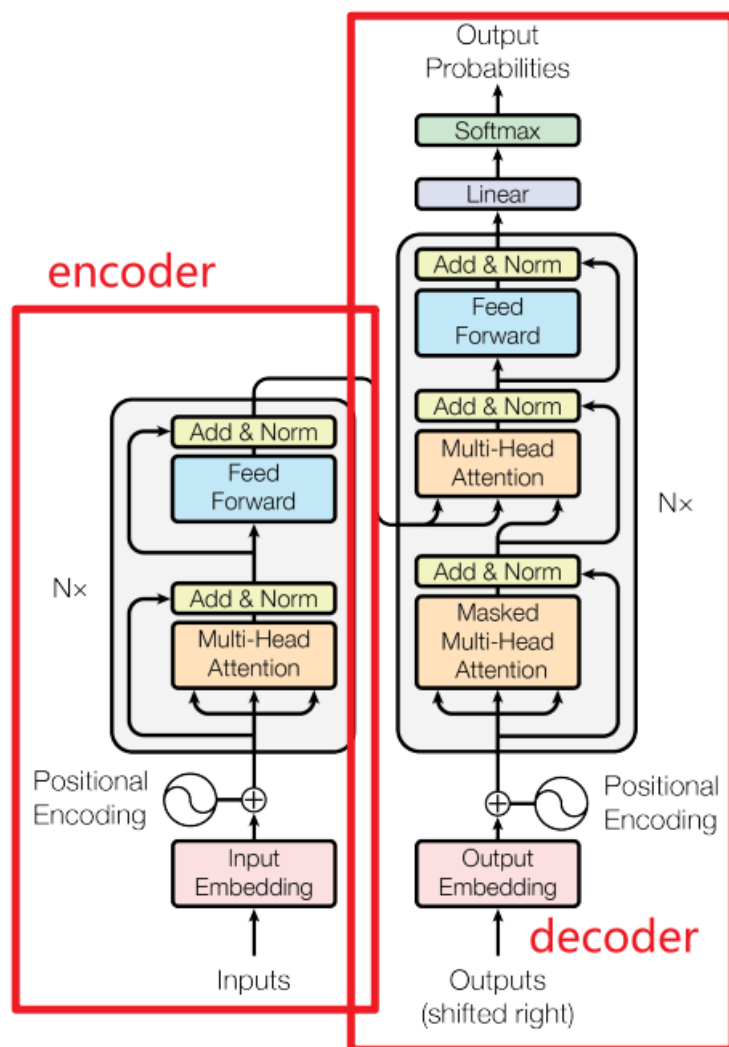


Figure 1: The Transformer - model architecture.

### 1.1.1. Encoder 编码器

- 由  $N = 6$  个相同的层构成，每层 (*layer*) 包含两个子层 (*sub-layer*)，第一个 *sub-layer* 叫做：  
*multi-head self-attention*，第二个 *sub-layer* 叫做：  
*simply position-wise fully connected feed-forward network*  
(其实就是 *MLP*，纯全连接层的神经网的模型)，每两个子层之间使用了残差连接，最后再使用 *layer normalization* 做层归一化以帮助梯度传播，残差需要输入和输出的大小是一样的，否则需要做投影，为了简单起见，把每一个层的输出维度变成 512 (即对每一个词不管在哪一层都做 512 的长度表示)。公式为：  
 $LayerNorm(x + Sublayer(x))$ ,  $d_{model} = 512$  (这样在调参的时

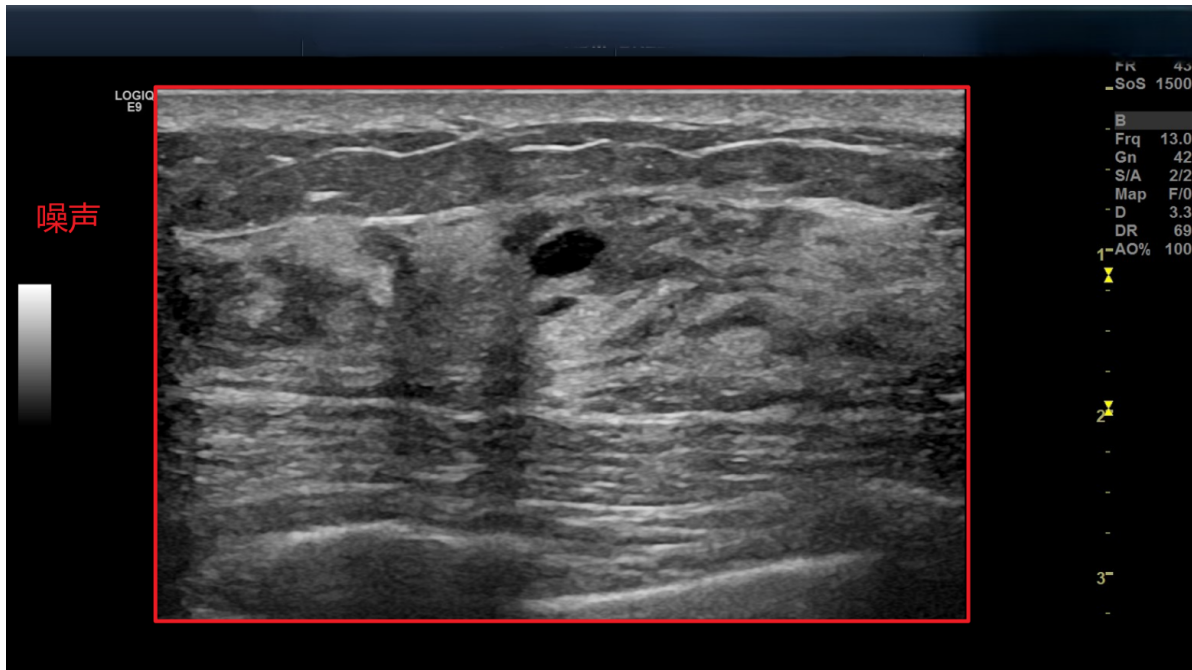
候就只用调两个参数，一个是输出维度 $d_{model}$ ，一个是层的个数 $N$ )

- 其中 *LayerNorm* (和 *batchNorm* 对比)：如果样本强度变化比较大，每次做小批量算出来的均值和方差的抖动是相对较大的；为了预测，需要把全局的均值和方差记录下来。如果全局的均值与方差碰到一个新的预测样本，这个样本非常长，此时之前算出来的均值和方差没有遇到过这么长的样本，就会导致预测结果不好使，而 *LayerNorm* 是基于每个样本自己来算均值和方差，更加稳定。简单说来：
  - *batchNorm* 是将每个特征归一化，*layerNorm* 是将每个样本归一化，
  - *NLP* 任务长度不固定，如果加入 0 值，*batchNorm* 会对这些 0 值一起归一化，所以用 *LayerNorm*

### 1.1.2. 解码器

- 同样是使用  $N = 6$  同样的层构成，不过由 3 个 *sub-layer* 构成，相较于编码器多出的第三个层依然使用 *Multi-Head Attention* (多头注意力机制)，和编码器一样，同样使用了残差连接。解码器做的是自回归，当前输出的输入集是上面一些时刻的输出，所以在做预测的时候不能看到本时刻之后的时刻的输出，但是在注意力机制里面，每一次都能看到整个完整的输入（在预测第 $t$ 个时刻的输出时不能看到 $t$ 时刻以后的输入），为了避免这个情况的发生，作者提出了带掩码的注意力机制 (*Masked Multi-Head Attention*) 在生成序列时，解码器通过关注编码器的输出和前面生成的部分来预测下一个 *token*。

## 1.2. 数据预处理



- 由于超声乳腺图像本身尺寸不一致、存在噪声以及不同成像设备之间的差异所带来的问题，我们主要进行了两种数据预处理操作：

### 1. Image\_Preprocess\_v1:

- **Resize to (256, 256):** 所有图像都被调整到固定的256x256大小
- **Center Crop to 224x224:** 从中心裁剪出224x224大小的图像，确保输入模型的图像大小一致
- **Convert to Tensor:** 将图像转换为PyTorch张量，方便模型处理
- **Normalize with mean and std of [0.5, 0.5, 0.5]:** 对图像进行归一化处理，将每个像素的RGB值缩放到[-1, 1]的范围。均值（mean）为0.5，标准差（std）为0.5

### 2. Image\_Preprocess\_v2:

- **Resize to (256, 256):** 同样调整所有图像到固定的256x256大小
- **Random Horizontal Flip (p=0.5):** 以50%的概率随机水平翻转图像，增加数据的多样性，避免模型对某一方向过拟合

- **Random Rotation (degrees=15)**: 随机旋转图像，最大旋转角度为15度，同样用于数据增强，增加训练数据的多样性
- **Color Jitter (brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1)**: 随机调整图像的亮度、对比度、饱和度和色调，让模型更具鲁棒性，防止过拟合到特定的光照或色彩条件
- **Center Crop to 224x224**: 从中心裁剪出224x224大小的图像，确保输入模型的图像大小一致
- **Convert to Tensor**: 将图像转换为PyTorch张量，方便模型处理
- **Normalize with mean and std of [0.5, 0.5, 0.5]**: 归一化操作，将图像的RGB值缩放到[-1, 1]的范围

## 1.3. 实现细节

1. 租赁云服务器参考硬件配置如下：

- 显卡类型：3090-24G
- CPU型号：Intel(R) Xeon(R) CPU E5-2682 v4
- 数据盘：200GB
- GPU数量：1

2. 实例镜像如下：

框架名称	框架版本	Cuda 名称	Python 名称
<b>PyTorch</b> >	2.2.1 >	11.6.0 >	3.8
TensorFlow >	2.0.0 >	<b>11.3.0</b> >	
PaddlePaddle >	1.13.1 >		
OneFlow >	1.13.0 >		
OpenCV >	<b>1.12.1</b> >		
MXNet >	1.12.0 >		
KataGo >	1.11.0 >		

请选择框架名称/框架版本/Cuda版本/Python版本 ^ 没有需要的镜像?

3. 进入云服务器开启终端，按requirements.txt或如下指令进行安装所需环境

```
1 # 手动清华源
2 pip config set global.index-url
  https://pypi.tuna.tsinghua.edu.cn/simple
3 # CUDA 11.3
4 conda install pytorch==1.12.1 torchvision==0.13.1
  torchaudio==0.12.1 cudatoolkit=11.3 -c pytorch &&
  pip install opencv-python
```

4. 调用run.py文件

## 1.4. 创新点

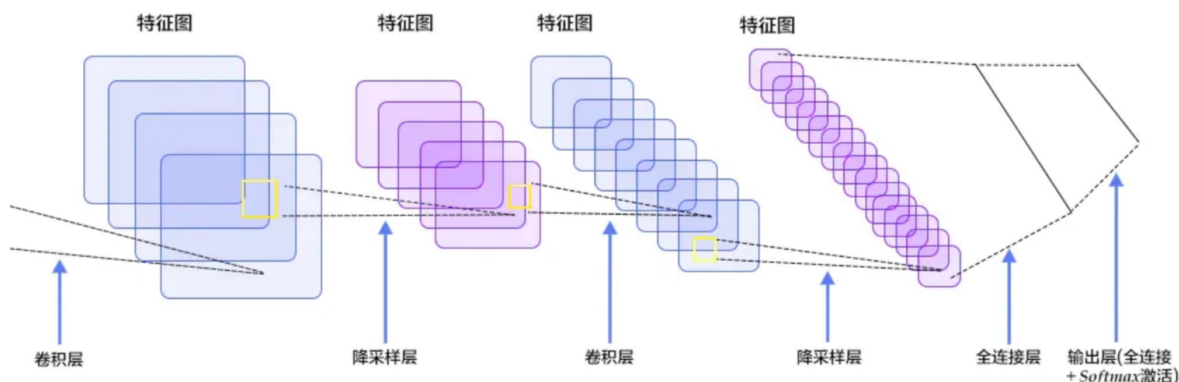
- **Encoder-Decoder架构**：使用6个相同的层构建编码器和解码器，采用多头自注意力机制和残差连接，增强了模型的表达能力和稳定性。在解码器中引入的掩码机制，使得生成过程自回归且具备时序一致性，提升了预测的准确性
- **LayerNorm的应用**：在模型中使用LayerNorm代替传统的BatchNorm，使得每个样本的归一化处理更加稳定，尤其适合处理长度不固定的序列数据，这在超声图像处理的多样性中表现尤为重要
- **多样化的数据增强策略**：引入随机翻转、旋转和颜色抖动等多种数据增强方法，使得训练数据更加丰富，显著提升了模型的鲁棒性与泛化能力

## 2. 特征识别

---

## 2.1 模型原理

- 使用以卷积神经网络（CNN）为主的模型，来识别乳腺影像中的BIRADS特征，包括边界、钙化、方向和形状。
- 识别过程如下所示：



## 2.2 数据预处理

- 在模型训练之前，我们对数据进行了预处理，以提高模型的性能和泛化能力。预处理步骤包括：
- **归一化**：将图像数据归一化到特定的均值和标准差，以减少模型训练中的内部协变量偏移。

```
1 val_transform = transforms.Compose([
2     transforms.ToTensor(),
3     transforms.Normalize(mean, std)
4 ])
```

- **二值化**：使用三角法对图像进行二值化处理，以突出乳腺病变的特征。

```
1 ret, thresh_triangle = cv2.threshold(gray, 0,
2     255, cv2.THRESH_BINARY + cv2.THRESH_TRIANGLE)
```



- **形态学操作：**通过形态学操作去除图像中的小的白点，以减少噪声对模型的影响。

```
1 kernel = np.ones((3, 3), np.uint8)
2 thresh_triangle =
  cv2.morphologyEx(thresh_triangle,
    cv2.MORPH_OPEN, kernel, iterations=2)
```

- **轮廓检测并裁剪：**使用OpenCV的轮廓检测功能来识别图像中最大的轮廓，这有助于定位乳腺病变区域。

```
1 contours, _ = cv2.findContours(thresh_triangle,
  cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
2 if contours:
3     largest_contour = max(contours,
  key=cv2.contourArea)
4     x, y, w, h =
  cv2.boundingRect(largest_contour)
5     cropped_img = img[y:y + h, x:x + w]
```

## 2.3 实现细节

模型的核心是一个多输出的神经网络，它基于MobileNetV2架构。MobileNetV2是一个轻量级的深度学习模型，它通过深度可分离卷积来减少计算量，同时保持较高的准确率。我们的模型通过以下步骤实现：

1. **特征提取：**使用MobileNetV2的前半部分作为特征提取器，提取输入图像的特征。这里，`self.base_model`是MobileNetV2模型的特征提取部分，它包含了卷积层和激活函数，用于从输入图像中提取特征。`last_channel`变量存储了最后一个卷积层的输出通道数，这个值将用于后续的全连接层的输入特征数。

```
1 self.base_model =  
  models.mobilenet_v2().features  
2 last_channel =  
  models.mobilenet_v2().last_channel
```

2. **特征融合**：通过自适应平均池化层（`AdaptiveAvgPool2d`），将MobileNetV2提取的特征图融合成一个固定大小（1x1）的向量。这一步骤将高维的特征图转换为一维的特征向量，为后续的分类器做好准备。

```
1 self.pool = nn.AdaptiveAvgPool2d((1, 1))
```

3. **分类器**：为每个BIRADS特征设计独立的分类器，这些分类器共享相同的特征向量，但具有独立的输出层。每个分类器都包含一个Dropout层和一个全连接层（`Linear`）。Dropout层有助于防止过拟合，而全连接层则负责将融合后的特征向量映射到每个BIRADS特征的类别空间。每个分类器都共享相同的输入特征向量，但具有独立的输出层，这样可以同时预测多个BIRADS特征。

```
1 self.boundary =  
    nn.Sequential(nn.Dropout(p=0.2),  
                  nn.Linear(in_features=last_channel,  
                             out_features=n_boundary_classes))  
2 self.calcification =  
    nn.Sequential(nn.Dropout(p=0.2),  
                  nn.Linear(in_features=last_channel,  
                             out_features=n_calcification_classes))  
3 self.direction =  
    nn.Sequential(nn.Dropout(p=0.2),  
                  nn.Linear(in_features=last_channel,  
                             out_features=n_direction_classes))  
4 self.shape = nn.Sequential(nn.Dropout(p=0.2),  
                             nn.Linear(in_features=last_channel,  
                                         out_features=n_shape_classes))
```

## 2.4 创新点

模型的创新之处在于：

- **多任务学习**：通过同时预测多个BIRADS特征，模型能够更全面地理解乳腺病变的特征，这有助于提高诊断的准确性。
- **轻量级架构**：使用MobileNetV2作为基础模型，使得我们的模型在保持高准确率的同时，具有较低的计算成本，适合在资源受限的环境中部署。

在实际应用中，我们通过一个自定义的函数 `characteristic` 来执行模型的预测。这个函数首先加载预训练的模型，然后对测试集中的图像进行预处理和预测，并将预测结果写入CSV文件。这个过程展示了模型在实际诊断流程中的应用，证明了其实用性和有效性。