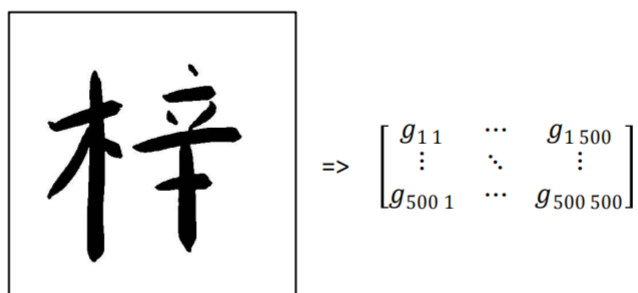


笔画分割概览

Target: 我们将笔画分割转化为多分类问题，输入一张 500×500 的二值化汉字图片，将不同类别的笔画分别抽取出来（笔画的分类将在文中提到），并提取出笔画的关键点（落笔点和提笔点）。我们希望模型能在拆解出笔画的同时，尽可能的保留笔画落笔点和提笔点（尤其是当笔画存在交叉时）的特征。

1. 图像特征定义

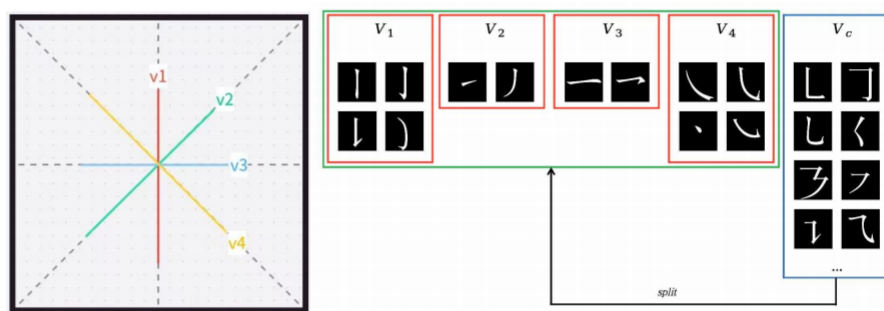
我们将原始的输入图像（二值化， 500×500 ）定义为 $G_{500 \times 500}$ 。



对于每个像素 g_{ij} ，我们为它设置 6 维特征 $[x_1, x_2, \dots, x_6]$ ，在我们讨论它们的定义之前，我们先对汉字中笔画的分类进行一个定义。

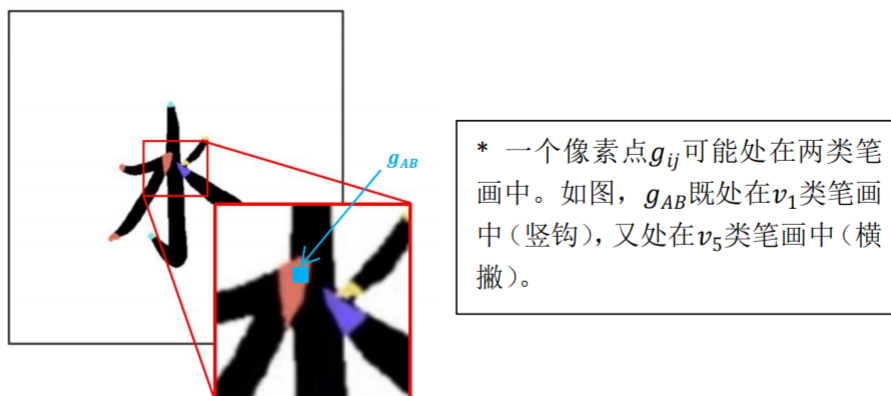
1.1 笔画分类

如图，笔画 v_1 至 v_4 按照“米”字格中四条线的方向按顺针对笔画进行分类，而 v_5 (v_c) 是那些较为复杂的笔画。



接下来回到每个像素 g_{ij} 的 6 维特征，对于 $k \in [1, 5]$ ， x_k 表示像素 g_{ij} 是否属于 v_k 类笔画，即：

$$x_{ijk} = \text{round}(P(g_{ij} \in G_{v_k})), \quad k = 1, 2, 3, 4, 5$$

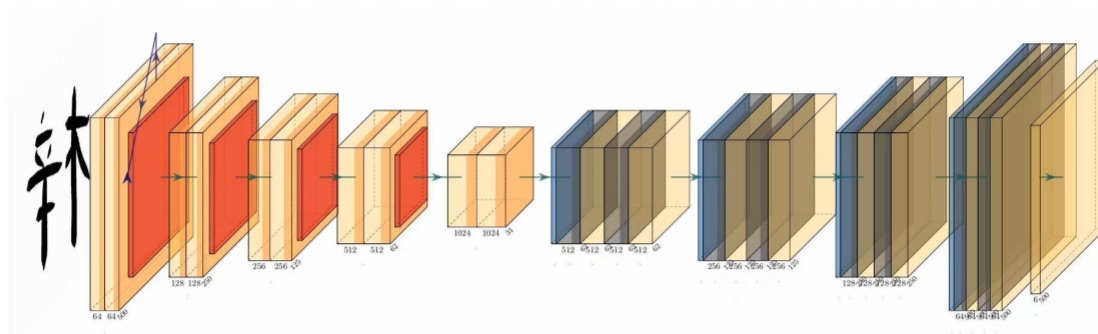


而对于 x_6 ，它表示像素点 g_{ij} 是否属于笔画的关键点（图中深绿色点）。



2. 模型训练

回到模型本身，我们的输入是一张 $500*500$ 的图片，输出是一个 $500*500*6$ 的张量，骨干模型采用 UNet，设置 4 层下采样和四层上采样。



我们将笔画分割问题转化为了多分类问题，因此在损失函数的选择上，我们采用了 BCE 函数。然而，为了让模型能够关注到一些更加重要的像素点，我们要为每个像素点的损失设置一些权重。

2.1 像素权重

对于每一个像素 g_{ij} ，我们定义其权重函数为 $W(i, j)$ ， $W(i, j)$ 需要满足以下性质：

- 1) 完全忽略掉白色的像素点，因为即使白色区域预测错误，我们也能以原始图像为掩码对白色区域进行修复。
- 2) 交叉点处的像素应该获得更高的权重，且在此处交叉的笔画越多，其权重越高。
- 3) 关键点处的像素应该获得更高的权重
- 4) 对于越接近边缘的像素点，它获得的权重应该越高，因为当笔画轮廓完整提取后，我们很容易将其内部填充完整。

基于以上性质，我们给出 $W(i, j)$ 的定义：

$$W(i, j) = g_{ij} \cdot (W_{feature}(i, j) + W_{contour}(i, j))^2$$

$W_{feature}(i, j)$ 用来满足性质 2 和性质 3，因为关键点和交叉点都已经被定义到了像素点的特征中，因此我们给出定义：

$$W_{feature}(i, j) = \sum_{k=1}^6 x_{ijk}$$

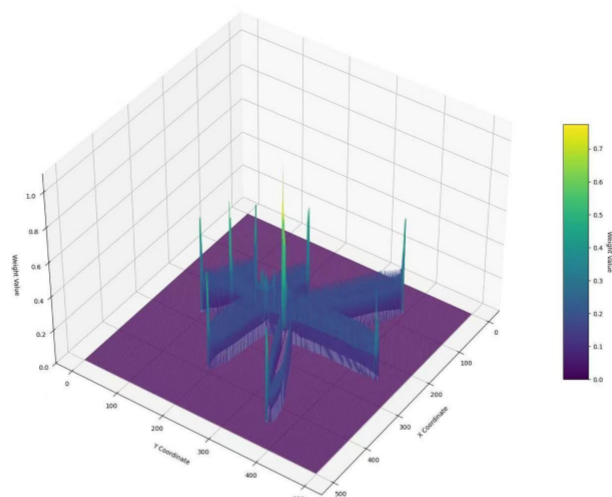
$W_{contour}(i, j)$ 为边缘权重，我们先给出定义：

$$W_{contour}(i, j) = (\min\{(g_{x,y} - g_{ij})^{-2} \cdot \sqrt{(x-i)^2 + (y-j)^2} \mid x \in (i-r, i+r), y \in (j-r, j+r)\})^{-1}, \\ x \neq i, y \neq j$$

我们使用了一个半径为 r （及边长为 $2r-1$ ）的矩形框作为窗口来探测像素点到边缘的最小距离，其中 $(g_{x,y} - g_{ij})^{-2}$ 用来确保我们计算的是当前像素点到边缘像素的距离，在找到最小距离后我们对其取倒数，使得离边缘越近，获得的权重越高。

最后，我们在 $W_{feature}(i, j) + W_{contour}(i, j)$ 前加上系数 g_{ij} 来使白色区域获得的权重为 0。为了防止总的 BCE loss 过大，我们再对 $W(i, j)$ 进行放缩使其标准化：

$$W_{scaled}(i, j) = \frac{W(i, j)}{\max(W(i, j)) - \min(W(i, j))}$$

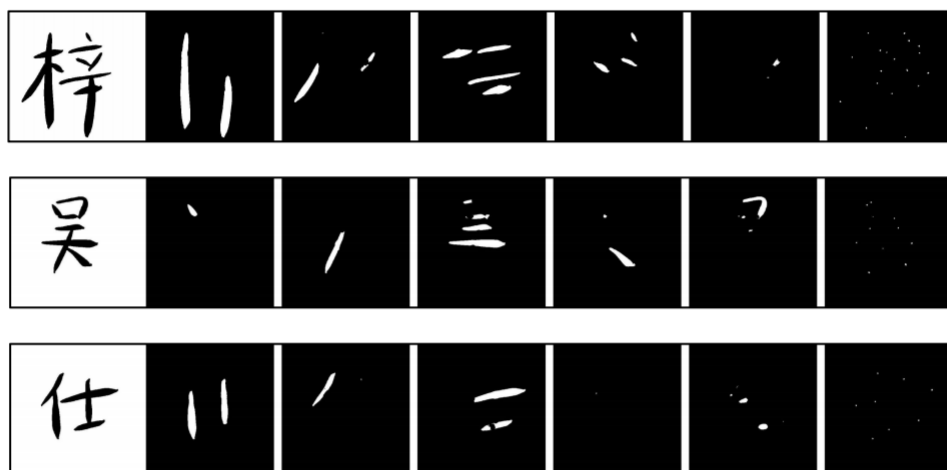


将 $W_{scaled}(i,j)$ 赋予 loss 函数，即：

$$global_loss = \text{mean}(W_{scaled} \cdot \text{BCE}(y, \hat{y}))$$

2.2 模型测试

我们目前基于人工收集的 400 张手写汉字数据集来验证理论可行性，其中 300 张作为训练集，100 张作为验证集，共进行 25 轮训练，在第 14 轮训练后训练损失和验证损失开始降低至稳定。目前模型在对于简单笔画字体能够有较好的表现：



3. 问题及解决方向

- 模型目前对于复杂笔画的提取能力明显不足，常将普通笔画错误的区分为复杂笔画，一方面 400 张数据集的规模过小，另一方面数据集内的字体没有包含生活中常见的复杂笔画，短期内目标将数据集扩充到 1200 张。

- 模型预测的结果并没有体现出其对边缘有足够高的关注度，因为当笔画中间被错误的截断时，没有看到模型有保留截断处边缘的迹象，推测原因有：

- 1) 边缘权重分配占比过小 ($W_{contour}(i, j) \in (0, 1]$ 而 $W_{feature}(i, j) \in [0, 6]$)
- 2) 边缘函数衰减过快：对于距离边缘为 2 及以上的点，其 $W_{contour}$ 会直接从 1 衰减到 0.5 以下。需要使用一个递减缓慢的函数来替代简单的对距离取倒数。

- 由于白色区域权重为 0，因此将权重放缩到(0,1)之间加到 BCE 损失上后会使全局损失在数值上过小。