

# 融合分类

2022年5月13日 9:03

# 融合分类

## 项目背景:

这项工作是给定了两种不同模态的遥感图像，一种是多光谱图像，另一种是全景图像。我们的任务是使用这两种模态的数据实现遥感图像的像素级分类。

批注 [李1]: MS 与 PAN 具体信息

## 数据介绍:

卫星传感器可以在同一区域内联合获取两种模态的遥感图像，即多光谱图像和全景图像。这两种形式的图像具有不同的特性，多光谱图像的光谱分辨率比较高，但空间分辨率较低，它有助于地物的识别。PAN 图像空间分辨率比较大，但光谱分辨率较低，他能够更准确的描述图像的的形状和轮廓信息。因此，我们需要实现两种模态数据的有效集成和利用，提高遥感图像的像素级分类任务精度。

## 基线问题:

起初我们使用 ResNet 作为特征提取器构建了一个双分支网络，分别对 MS 和 PAN 提取特征，融合双分支特征之后进行最终的地物分类，但分类效果并不理想。原因是没有考虑 PAN 和 MS 图像之间的模态差异，使得融合特征的辨别能力没有那么高。

批注 [李2]: ResNet 原理

## 两个痛点:

对于这项任务，我们需要解决两个主要问题，一个是 PAN 和 MS 图像的表象差异，这是由于两种图像数据来自不同的成像传感器造成的。第二个问题是，由于两种模态数据的倾向性不同，他们提取的高层语义特征也会存在一定的偏差。

### 痛点问题 1 解决方案:

针对第一个问题，我们在特征提取之前对 PAN 和 MS 图像进行了预处理，即采用了一种双向的优势特征监督机制。我们利用两个模态的优势特征建立了一个联合优化目标函数，利用一个模态的优势特征作为监督信息逐步引导另一个模态特征的生成，从而自适应地添加彼此的优势信息，以减少 PAN 和 MS 图像之间的模态差异。我们分为两个部分，一个是利用 MS 的光谱信息作为监督信息来指导 PAN 的特征生成，另一部分则是通过 PAN 的空间信息作为监督信息来指导 MS 特征的生成。

#### 实施细节 1:

这里，我以利用 MS 的光谱信息作为监督信息牵引 PAN 的特征生成为例向您具体说明一下具体的实施细节。我们首先利用 GIHS 颜色转换算法将 MS 图像分离为强度 I，色调 H 和饱和度 S 三个分量，I 表示了光谱的整体亮度，代表了空间信息，而 H 和 S 都与光谱的波长有关，他们表示了光谱分辨率。其次，我们受到矩阵相似性的启发，即如果两个矩阵 A 和 B 相似，那么存在矩阵 P，有  $P^{-1}AP=B$ 。我们利用卷积在最后一层 PAN 特征图上计算这个转换矩阵，然后根据上面的等式建立了优化函数。同理，我们也可以利用 PAN 的优势特征 I 分量与从 MS 的特征图中计算的转换矩阵建立优化函数。然后对这部分网络先进行训练，使得网络提取的两种模态的特征差异逐渐减小。

批注 [李3]: HIS 原理

批注 [李4]: L1 Loss 等

### 痛点问题 2 解决方案:

针对高层语义偏差的问题，我们采用了一种交互式的相关特征匹配方法。我们通过计算不同模态之间的相互依赖性来评估多模态特征的内部相关性，然后从空间和光谱两个维度逐层对多模态特征进行相关匹配。强调单个模态的强依赖性特征，从而逐渐减少模态之间的语义特征偏差。

**实施细节 2:**

在具体实现的时候，前面也已经提到了，对于成对的 MS 和 PAN 图像，我们是从光谱和空间两个维度来增强他们的强相关特征的。

首先是从光谱维度，我们通过计算 MS 和 PAN 之间的互信息（单模态信息熵和多模态联合熵），来获取 MS 特征的每个通道与 PAN 图的相关性度量值的，这也相当于一种注意力机制，通过互信息，我们获得一个通道掩膜，将他作用于 MS 特征，实现对强相关的 MS 特征的强调。

而在空间维度，我们首先计算了 MS 和 PAN 图像之间的相似性（定义公式），最终的相似性是以一个二维矩阵的方式存在的，我们对矩阵的每一行求最大值，就可以找到与 MS 图像强相关的 PAN 图通道，选择与 MS 相关性最高的 topK 个 PAN 图通道，将这些通道与之前求得的相关性系数加权求和就获得了最终的空间掩膜，将空间掩膜作用于 PAN 图，强调与 MS 强相关的 PAN 图特征。

**最终结果:**

最后，我们将每一层的单模态特征级联在一起，再将级联之后的两种模态特征元素级相加，得到融合特征，最后利用全连接层实现分类。最终我们取得了 92% 的分类精度，较基线模型提高了 5%。

**优化:**

我们的训练是分为两个阶段完成的，之后的话，希望能在数据预处理的部分在深入思考一下，尽量形成端到端的框架。

批注 [李5]: 互信息

批注 [李6]: 信息熵

批注 [李7]: 相似性度量方式

批注 [李8]: 全连接层，PyTorch 全连接层实现

## • PAN和MS影像信息

### • 数据源信息:

① 二为二色, 光谱分辨率为 $0.8m$ , 所利用的波段为R, G, B以及近红外波段

② 光谱信息更加丰富, 更有利用反映植物内部特征。

### • 全色图像

• 全色图像是对地物中两种中全色波段影像的提取(可见光波段)

• 全色波段空间分辨率为 $0.2m$ , 更有利于反映地物形状、尺寸、边缘信息。

### • 二为二色图像与全色图像的区别:

① 二为二色图像的波段更多, 图像为波段可以捕获到更多个非常窄的波段信息。

② 二为二色图像的波段较少(10个左右)

③ 与MS相比, 二为二色信息更加丰富。

## • RGB 颜色转换

• 思路: RGB 图像有效高为 3 个分量: 红度, 绿度, 蓝色度。

红度: 反映红光的整体亮度, 代表着空间信息。

色度, 饱和度: H 和 S 都与光的波长有关, 因此共同表示了颜色信息。

I, H, S 很独立, 可以分别进行处理。

## • 三通道亮度处理:

$$I = \frac{R+G+B}{3} \quad S = \frac{1-\min(R,G,B)}{R+G+B}$$

$$\theta = \arccos \frac{0.5 \times [(R-G) + (R-B)]}{\sqrt{(R-G)^2 + (R-B)(G-B)}}$$

$$H = \begin{cases} \theta & B \leq G \\ 360 - \theta & B > G \end{cases}$$

## • GEMS:

$$I = \frac{R+2G+B+N}{4} \quad (G, B \text{ 都不是零且包含在 } [0, 255] \text{ 范围内})$$

x, y, z 分量 → 红 近红外 中红外

## • L1 Loss 等

L1 loss 在零上不平滑, 不常使用 常用于简单的回归任务

### • L1 Loss:

$$\text{loss}(x, y) = \frac{1}{n} \sum_{i=1}^n |y_i - f(x_i)|$$

L1 loss 对异常点比较敏感, 需要手动调整权重, 避免梯度爆炸  
适合回归任务, 数据噪声不大, 问题线性度不高

### • L2 loss

$$\text{loss}(x, y) = \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2$$

### • Smooth L1 loss

$$\text{loss}(x, y) = \frac{1}{n} \sum_{i=1}^n \begin{cases} 0.5 (y_i - f(x_i))^2 & |y_i - f(x_i)| < 1 \\ |y_i - f(x_i)| - 0.5 & \text{otherwise} \end{cases}$$



Smooth L1 loss 解决零上不平滑问题, 对异常值更加鲁棒, 损失函数更加平滑, 不易对异常值过度拟合

## • L1, L2 正则化

### • L1 正则化:

$$\sum (y_i - w^T x_i)^2 + \lambda \sum |w_j| \quad (\lambda \text{ 为正则化系数之和})$$

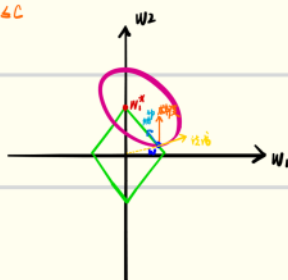
L1 使模型参数稀疏, 由于参数稀疏, 过滤掉一些无用的特征, 防止过拟合

### • L2 正则化:

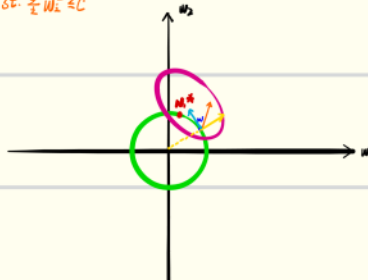
$$\sum (y_i - w^T x_i)^2 + \lambda \sum w_j^2$$

L2 使模型参数更小, 即平滑参数, 从而防止过拟合

min  $J(w, y)$   
st:  $\sum |w_j| \leq C$



min  $J(w, y)$   
st:  $\sum w_j^2 \leq C$



## • 对特征未用 L2 正则

• 作用: ①防止梯度消失或爆炸

②给一量纲, 加速模型收敛

(例如 SSD 的 conv4-3 层加了 L2 norm)

当模型训练几个 batch 后, loss 会变成 nan, 在特征后加 L2 正则可以解决

• L2 正则原理:  $y = \frac{x_i}{\sum x_i^2}$  为 X 向量长度

## • pyTorch 支持:

`torch.nn.L1Loss()`

`torch.nn.L2Loss()`

`torch.nn.SmoothL1Loss()`

## • 互信息, 信息熵

• 互信息: 互信息可以看作是一个随机变量中包含的关于另一种随机变量的信息量, 反映了两种变量之间的相互依赖程度。

互信息的极值是无穷。

$$I(X, Y) = H(X) + H(Y) - H(X, Y)$$

$$I(X, Y) = \sum p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$

• 熵: 熵描述了系统的混乱程度 (不确定性)

对于每一事件来说, 如果它出现所有状态的概率相等, 那么, 此时熵有最大, 此时系统的选择没有偏向性, 混乱程度最大。

• 自信息:  $I(x) = -\log(p(x))$  事件的概率越小, 自信息越大, 自信息只能对单个变量的信息量进行衡量。

• 信息熵: 信息熵代表了事件的信息量, 依赖于事件的概率分布  $p(x)$  信息熵是对整个系统的确定性进行衡量

$$H(X) = -\sum p(x) \log(p(x))$$

对于图像来说, 将图像的值归则分到 [0, 255],  $p(x)$  表示图像中  $pixel = x$  的概率。

- **联合熵**: 描述了两个变量同时发生时的信息熵。

$$H(X, Y) = -\sum_{x,y} p(x,y) \log(p(x,y))$$

KL散度:

- **相对熵**: 衡量两个概率分布之间的差异。(同一个变量的两个概率分布)

$$D_{KL}(p||q) = -\sum p(x) \log\left(\frac{p(x)}{q(x)}\right)$$

KL值小, 两分布越接近。

- **交叉熵**: 用于衡量在给定真实分布  $p(x)$  的条件下, 利用非真实分布  $q(x)$  所制定策略来减少系统的不确定性所付出的努力的大小。

利用最接近真实分布的策略来降低系统的不确定性, 所要付出的努力肯定是最小的。

$$H(p, q) = -\sum p(x) \log(q(x))$$

- **信息熵**, **相对熵**, **交叉熵**的关系。

$$D_{KL}(p||q) = -\sum p(x) \log\left(\frac{p(x)}{q(x)}\right) = -\left(\sum p(x) \log(p(x)) - \sum p(x) \log(q(x))\right) = H(p, q) - H(X)$$

相对熵 = 某个策略的交叉熵 - 信息熵。



- 相似性度量方法

- 余弦: 
$$\delta = \frac{(R_i^M - \bar{R}^M)^T (R_i^L - \bar{R}^L)}{\|R_i^M - \bar{R}^M\|_2 \cdot \|R_i^L - \bar{R}^L\|_2}$$
 (衡量两资源之间的整体相似性)  $\delta \in [-1, 1]$   

$$= \frac{(R_i^M - \bar{R}^M) \cdot (R_i^L - \bar{R}^L)}{\sqrt{(R_i^M - \bar{R}^M)^2} \cdot \sqrt{(R_i^L - \bar{R}^L)^2}}$$
 当  $(R_i^M - \bar{R}^M)$  与  $(R_i^L - \bar{R}^L)$  的变化趋势相同时,  $\delta \rightarrow 1$ , 即两者越相似,  $\delta$  越大.

- 欧氏距离:  $d = \sqrt{\sum (x_i - y_i)^2}$

- 曼哈顿距离:  $d = \sum |x_i - y_i|$

- 明可夫斯基距离:  $d = \sqrt[p]{\sum |x_i - y_i|^p}$

- 夹角余弦:  $\cos \theta = \frac{X \cdot Y}{\sqrt{X^2} \cdot \sqrt{Y^2}}$   $X \cdot Y = |A| \cdot |B| \cdot \cos \theta$

- 杰卡德相似系数:  $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$

- 皮尔逊相关系数:  $\rho_{XY} = \frac{\text{cov}(X, Y)}{\sqrt{\text{D}(X)} \cdot \sqrt{\text{D}(Y)}} = \frac{E[(X - E(X))(Y - E(Y))]}{\sqrt{\text{D}(X)} \cdot \sqrt{\text{D}(Y)}}$

$$|\rho_{XY}| \leq 1, \quad \rho_{XY} \rightarrow 1, \text{ 正相关} \quad \rho_{XY} \rightarrow -1, \text{ 负相关} \quad |\rho_{XY}| \rightarrow 0, \text{ 相关性降低, } 0 \text{ 不相关}$$

- 全连接层

网子为类 `torch.nn.Linear(M, class)`  $M = CXHXW$  (除去 Batch Size 维度)



# 融合分类项目问题

2022年7月16日 17:08

## 1. 研究生比赛

- Kaggle的一个新冠肺炎检测比赛
- 根据人体肺部影像中的一些斑驳去判断是否患有新冠
- 用CenterNet做的，但没有拿到好的成绩
- 样本差异太小，采用了Triplet loss进行改进

## 2. 为什么不用语义分割的方式

- 也可以看做是一种语义分割模型，但传统的语义分割的实现方法会有一些不同
- 具体问题：对于位置很近但类别不同的像素点，容易出现误分类，需要相关性强的邻域信息
  - 传统的语义分割模型
    - 一般是通过特征粒度的处理（多尺度卷积并行处理等）
    - 对于位置相邻，但类别不同的像素点来说，提取的邻域信息相似度高
  - OURS
    - 我们以每个待预测像素点为单位，生成一对Patch送入网络
    - 在Patch的生成过程中，利用不同的采样策略，我们就可以使它是包含与待分类像素点强相关的邻域信息
      - ◆ 中心像素点偏移策略
        - ◇ 利用灰度共生矩阵计算待预测像素点水平和垂直四个方向上的像素信息相关性
        - ◇ 根据相关性确定偏移方向和距离，提取中心偏移后的邻域信息输入网络进行计算
        - ◇ 如果四个方向的相关性都大于设定阈值，那么证明像素点位于平坦区域，不需要迁移
      - ◆ 处理方式上会更加精细一点

## 3. Kappa一致性检验

- Kappa=1：说明两个结果完全一致
- Kappa=-1：说明两个结果完全不一致
- Kappa=0.75-1：说明一致性较好
- Kappa=0.4-0.75：说明一致性中等
- Kappa=0-0.4：一致性很差
- Kappa=0：说明结果具有随机性
- 计算

混淆矩阵：

成绩评级 好 中 差

好	a	b	c
中	d	e	f
差	g	h	i

学生实际的成绩评级和预测的成绩评级

$$p_o = \frac{a+e+i}{a+b+c+d+e+f+g+h+i}$$

$$p_e = \frac{(a+d+g) \times (a+b+c) + (b+e+h) \times (d+e+f) + (c+f+i) \times (g+h+i)}{(a+b+c+d+e+f+g+h+i)^2}$$

将  $p_o$  和  $p_e$  代入  $kappa = \frac{p_o - p_e}{1 - p_e}$  即可。

- 即所有类别分别对应的“实际数量与预测数量的乘积” / “样本总数的平方”
- 越不平衡的混淆矩阵， $p_e$ 越高