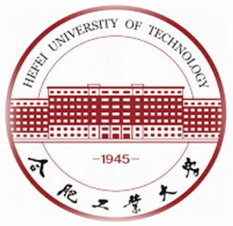
Rendezvous: Attention Mechanisms for the Recognition of Surgical Action Triplets in Endoscopic Videos



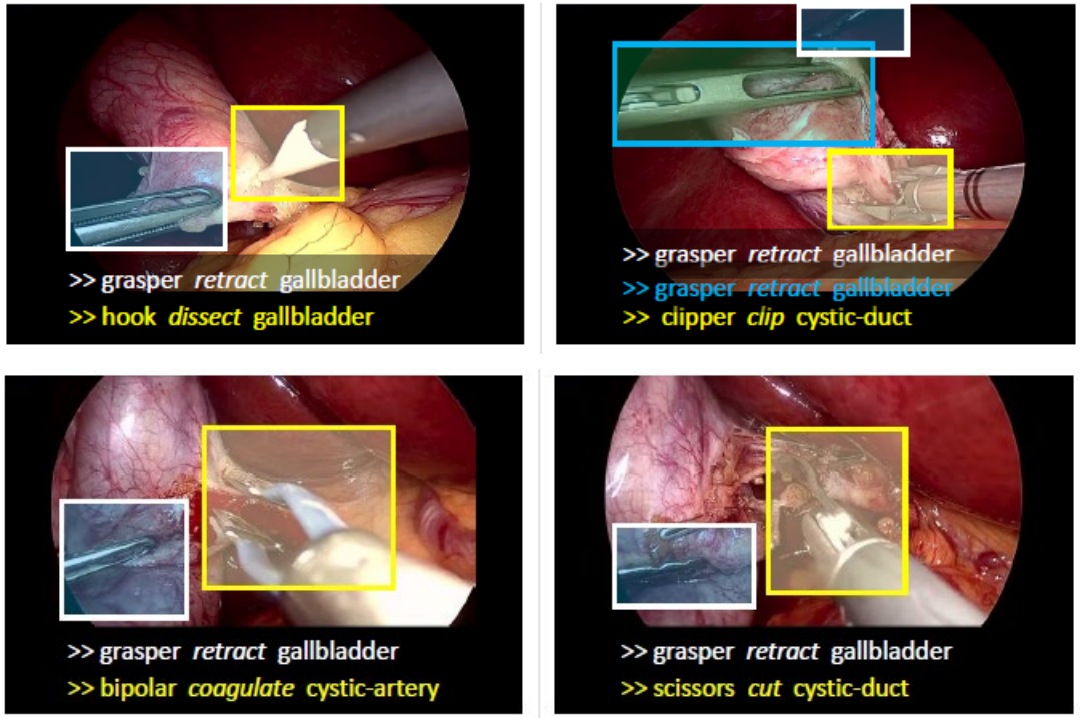
合肥工業大學

苏伊阳

2022.9.23

研究背景

手术三元组识别：<器械，动作，目标部位>，包括识别手术器械、手术动作、手术目标部位以及三者之间的关系。识别手术三元组，可以更加全面的反应手术场景的信息，提高手术安全性和效率。



[1] Nwoye C I, Yu T, Gonzalez C, et al. Rendezvous: Attention mechanisms for the recognition of surgical action triplets in endoscopic videos[J]. Medical Image Analysis, 2022, 78: 102433.

研究背景

挑战：

- **三元组以手术器械为中心：手术动作和手术目标部位高度依赖于手术器械，例如，在胆囊腹腔镜切除术中大部分时间都能看到肝脏，但只有在被器械作用时才被认为是目标部位。同样，动作也是由器械决定的，没有器械，就没有动作。**
- **空间推理（spatial reasoning）：对于同一个器械和目标部位，可能会有不同的手术动作：<grasper, retract, gallbladder>, <grasper, grasp, gallbladder>, <grasper, dissect, gallbladder>（缩回；抓握；解剖），对医生来说都很难区分是什么动作**
- **多重性和语义推理：同一个手术动作会出现在不同的手术器械中**

<bipolar, dissect, gallbladder>, <grasper, dissect, gallbladder>;

一个目标部位会同时出现不同的器械和动作

<grasper, retract, cystic-duct>， <hook, dissect, cystic-duct>

[1] Nwoye C I, Yu T, Gonzalez C, et al. Rendezvous: Attention mechanisms for the recognition of surgical action triplets in endoscopic videos[J]. Medical Image Analysis, 2022, 78: 102433.

研究背景

Table 2: Dataset statistics showing the number of occurrences of the triplets

Name	Count	Name	Count	Name	Count
bipolar,coagulate,abdominal-wall/cavity	434	grasper,grasp,cystic-artery	76	hook,dissect,gallbladder	29292
bipolar,coagulate,blood-vessel	251	grasper,grasp,cystic-duct	560	hook,dissect,omentum	3649
bipolar,coagulate,cystic-artery	68	grasper,grasp,cystic-pedicle	26	hook,dissect,peritoneum	337
bipolar,coagulate,cystic-duct	56	grasper,grasp,cystic-plate	163	hook,null-verb,null-target	4397
bipolar,coagulate,cystic-pedicle	77	grasper,grasp,gallbladder	7381	hook,retract,gallbladder	479
bipolar,coagulate,cystic-plate	410	grasper,grasp,gut	33	hook,retract,liver	179
bipolar,coagulate,gallbladder	343	grasper,grasp,liver	83	irrigator,aspirate,fluid	3122
bipolar,coagulate,liver	2595	grasper,grasp,omentum	207	irrigator,dissect,cystic-duct	41
bipolar,coagulate,omentum	262	grasper,grasp,peritoneum	380	irrigator,dissect,cystic-pedicle	89
bipolar,coagulate,peritoneum	73	grasper,grasp,specimen-bag	6834	irrigator,dissect,cystic-plate	10
bipolar,dissect,adhesion	73	grasper,null-verb,null-target	4759	irrigator,dissect,gallbladder	29
bipolar,dissect,cystic-artery	187	grasper,pack,gallbladder	328	irrigator,dissect,omentum	100
bipolar,dissect,cystic-duct	183	grasper,retract,cystic-duct	469	irrigator,irrigate,abdominal-wall/cavity	413
bipolar,dissect,cystic-plate	54	grasper,retract,cystic-pedicle	41	irrigator,irrigate,cystic-pedicle	29
bipolar,dissect,gallbladder	353	grasper,retract,cystic-plate	1205	irrigator,irrigate,liver	130
bipolar,dissect,omentum	176	grasper,retract,gallbladder	48628	irrigator,null-verb,null-target	573
bipolar,grasp,cystic-plate	8	grasper,retract,gut	686	irrigator,retract,gallbladder	30
bipolar,grasp,liver	95	grasper,retract,liver	13646	irrigator,retract,liver	350
bipolar,grasp,specimen-bag	85	grasper,retract,omentum	4422	irrigator,retract,omentum	89
bipolar,null-verb,null-target	632	grasper,retract,peritoneum	289	scissors,coagulate,omentum	17
bipolar,retract,cystic-duct	8	hook,coagulate,blood-vessel	57	scissors,cut,adhesion	155
bipolar,retract,cystic-pedicle	9	hook,coagulate,cystic-artery	10	scissors,cut,blood-vessel	21
bipolar,retract,gallbladder	32	hook,coagulate,cystic-duct	41	scissors,cut,cystic-artery	613
bipolar,retract,liver	164	hook,coagulate,cystic-pedicle	15	scissors,cut,cystic-duct	808
bipolar,retract,omentum	69	hook,coagulate,cystic-plate	9	scissors,cut,cystic-plate	20
clipper,clip,blood-vessel	51	hook,coagulate,gallbladder	217	scissors,cut,liver	90
clipper,clip,cystic-artery	1097	hook,coagulate,liver	189	scissors,cut,omentum	27
clipper,clip,cystic-duct	1856	hook,coagulate,omentum	78	scissors,cut,peritoneum	56
clipper,clip,cystic-pedicle	13	hook,cut,blood-vessel	15	scissors,dissect,cystic-plate	12
clipper,clip,cystic-plate	53	hook,cut,peritoneum	92	scissors,dissect,gallbladder	52
clipper,null-verb,null-target	309	hook,dissect,blood-vessel	21	scissors,dissect,omentum	93
grasper,dissect,cystic-plate	78	hook,dissect,cystic-artery	2984	scissors,null-verb,null-target	171
grasper,dissect,gallbladder	644	hook,dissect,cystic-duct	7861		
grasper,dissect,omentum	31	hook,dissect,cystic-plate	2898	Total	161005

研究方法

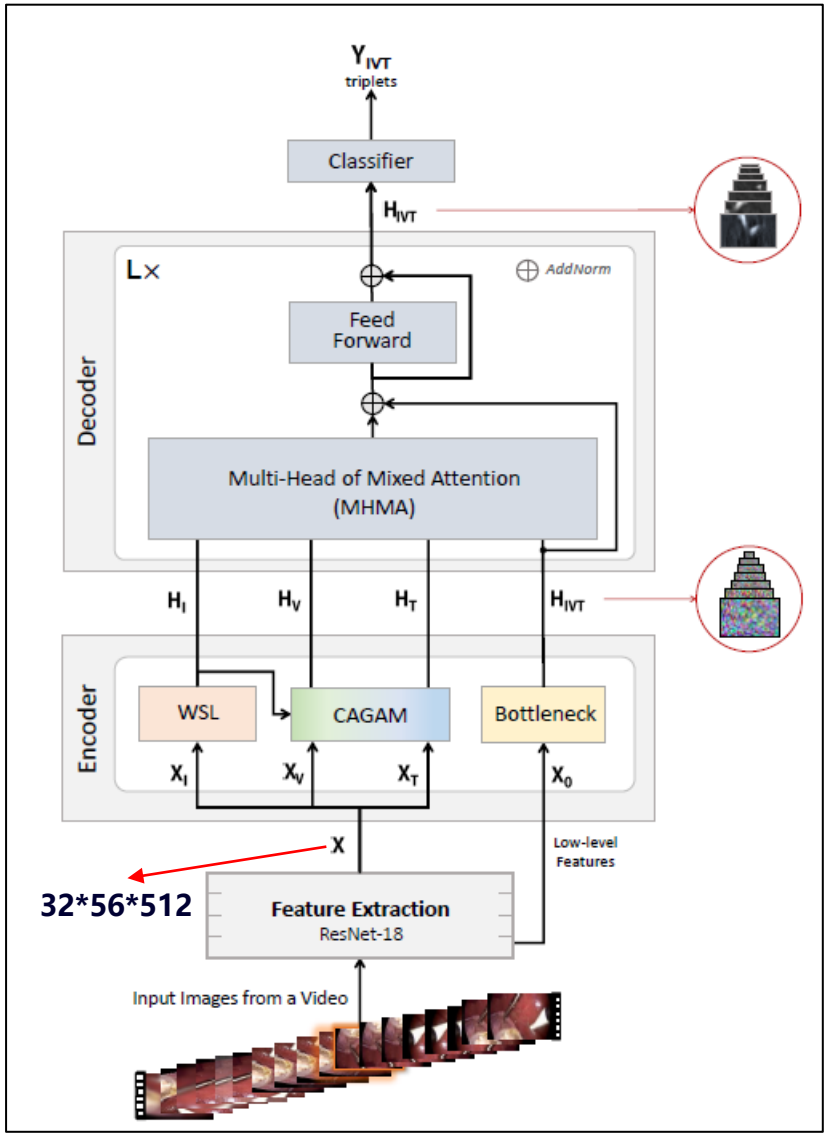
由编码器和解码器组成，编码器负责检测三元组的各个组成部分，解码器则解决它们之间的关系。

- 编码器：弱监督定位(WSL), CAGAM, Bottleneck
- 解码器：多头混合注意（MHMA)

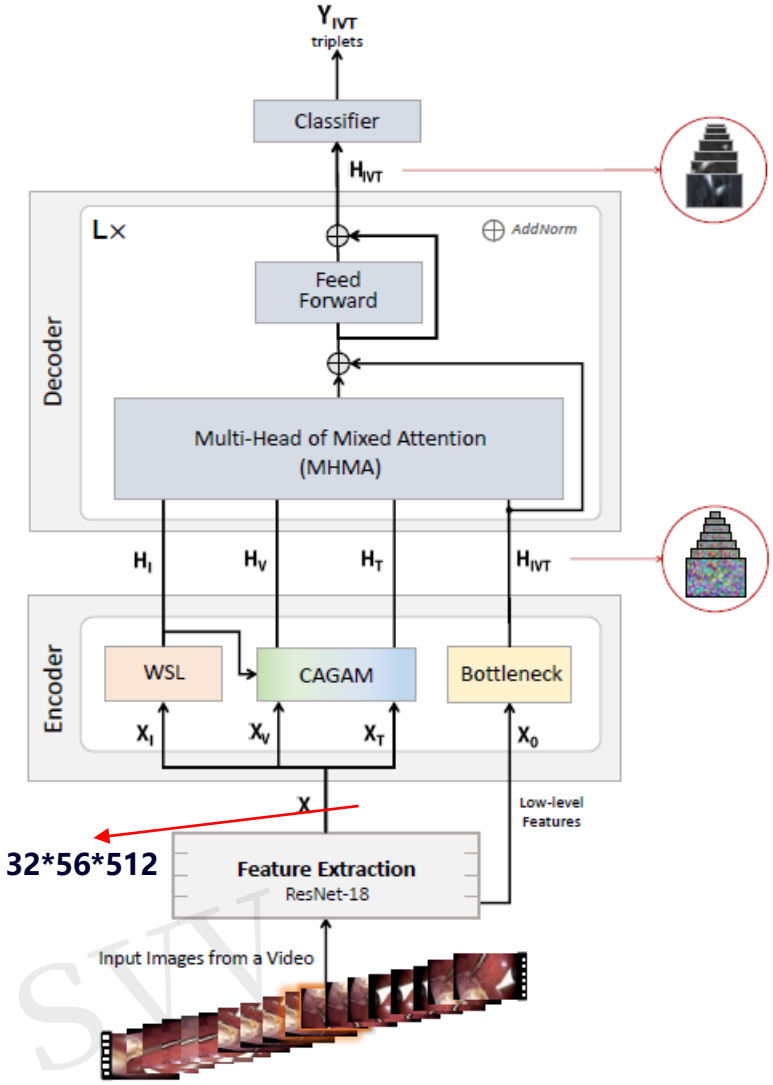
弱监督定位(WSL)模块：用于手术器械检测的

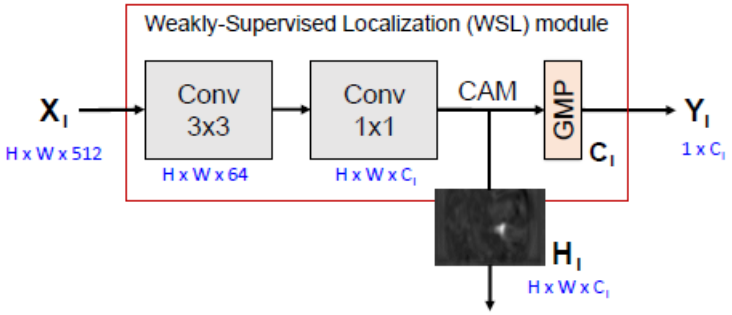
类激活引导注意 (CAGAM)模块：用于动作和目标识别

Bottleneck模块：从Resnet-18收集未过滤的底层特征（第一个block），提供了三元组的全局特征



研究方法



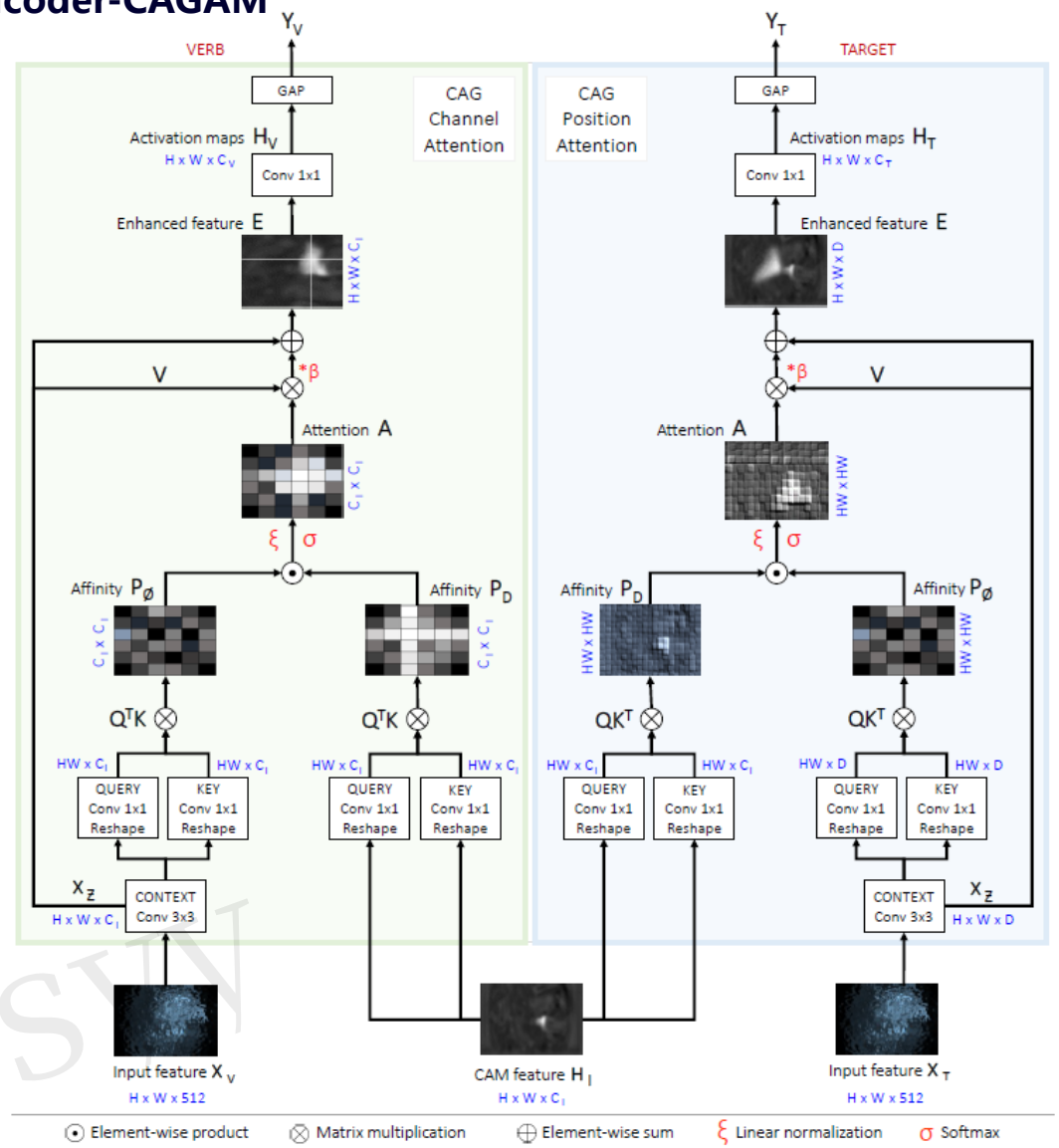



```
class WSL(nn.Module):
    def __init__(self, num_class, depth=64):
        super(WSL, self).__init__()
        self.conv1 = nn.Conv2d(in_channels=512, out_channels=depth, kernel_size=3, padding=1)
        self.cam = nn.Conv2d(in_channels=depth, out_channels=num_class, kernel_size=1)
        self.elu = nn.ELU()
        self.bn = nn.BatchNorm2d(depth)  # channel维度标准化
        self.gmp = nn.AdaptiveMaxPool2d((1, 1))

    def forward(self, x):
        feature = self.conv1(x)
        feature = self.bn(feature)
        feature = self.elu(feature)
        cam = self.cam(feature)
        logits = self.gmp(cam).squeeze(-1).squeeze(-1)
        return cam, logits
```

研究方法

Encoder-CAGAM



CAGAM模块：由channel attention和position attention组成。

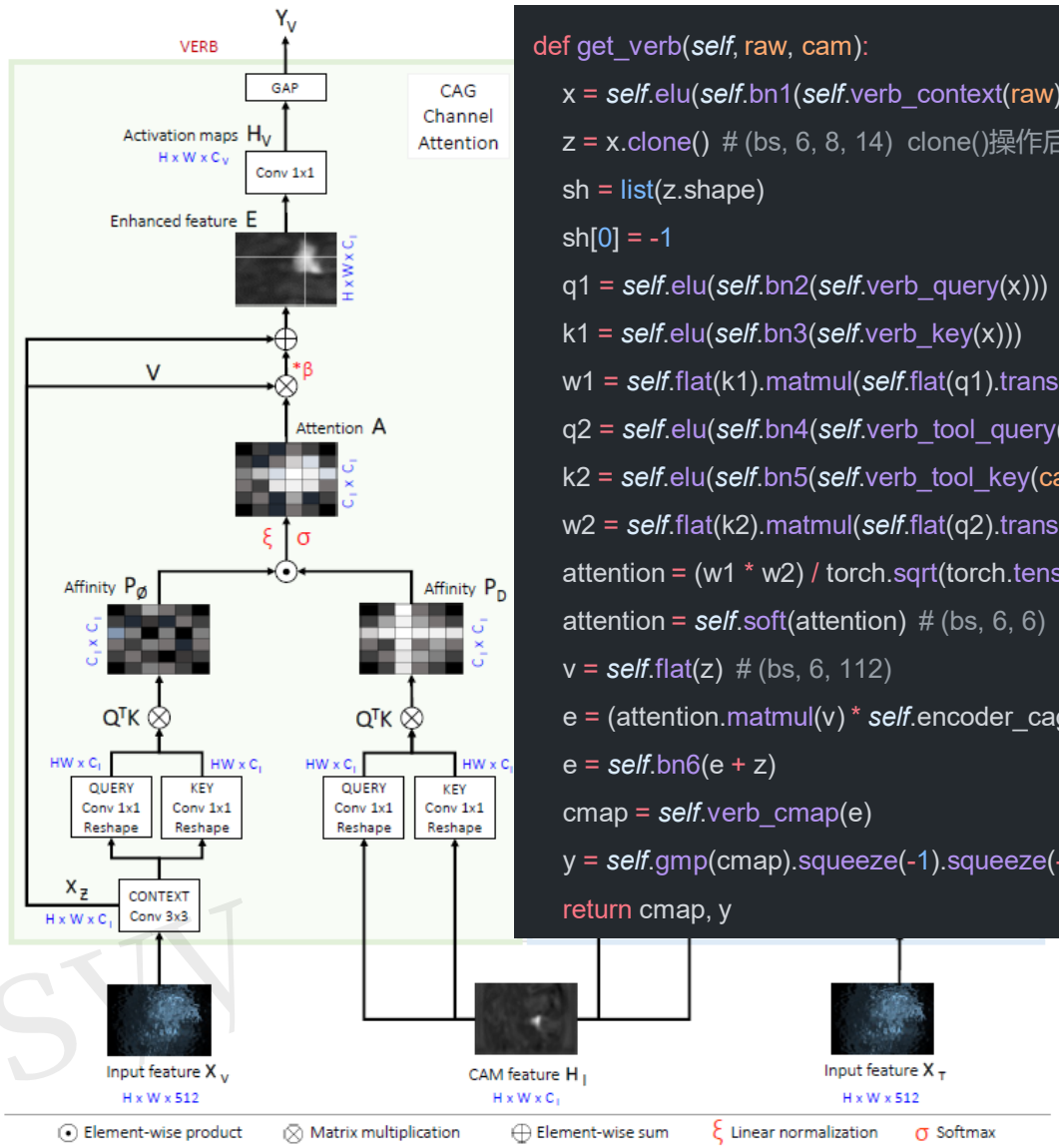
作者认为动作和目标部位针对特定器械表现不同，动作更容易受到器械类型的影响，目标部位更容易受到器械位置的影响。

channel attention（verb）：通道注意在$C_I * C_I$特征图中捕获，可以表示器械的类型

position attention（target）：位置注意在$HW * HW$特征图中捕获，表示器械的空间位置

通过Element-wise product进行特征融合，用来抑制或突出某个区域内的特性

研究方法



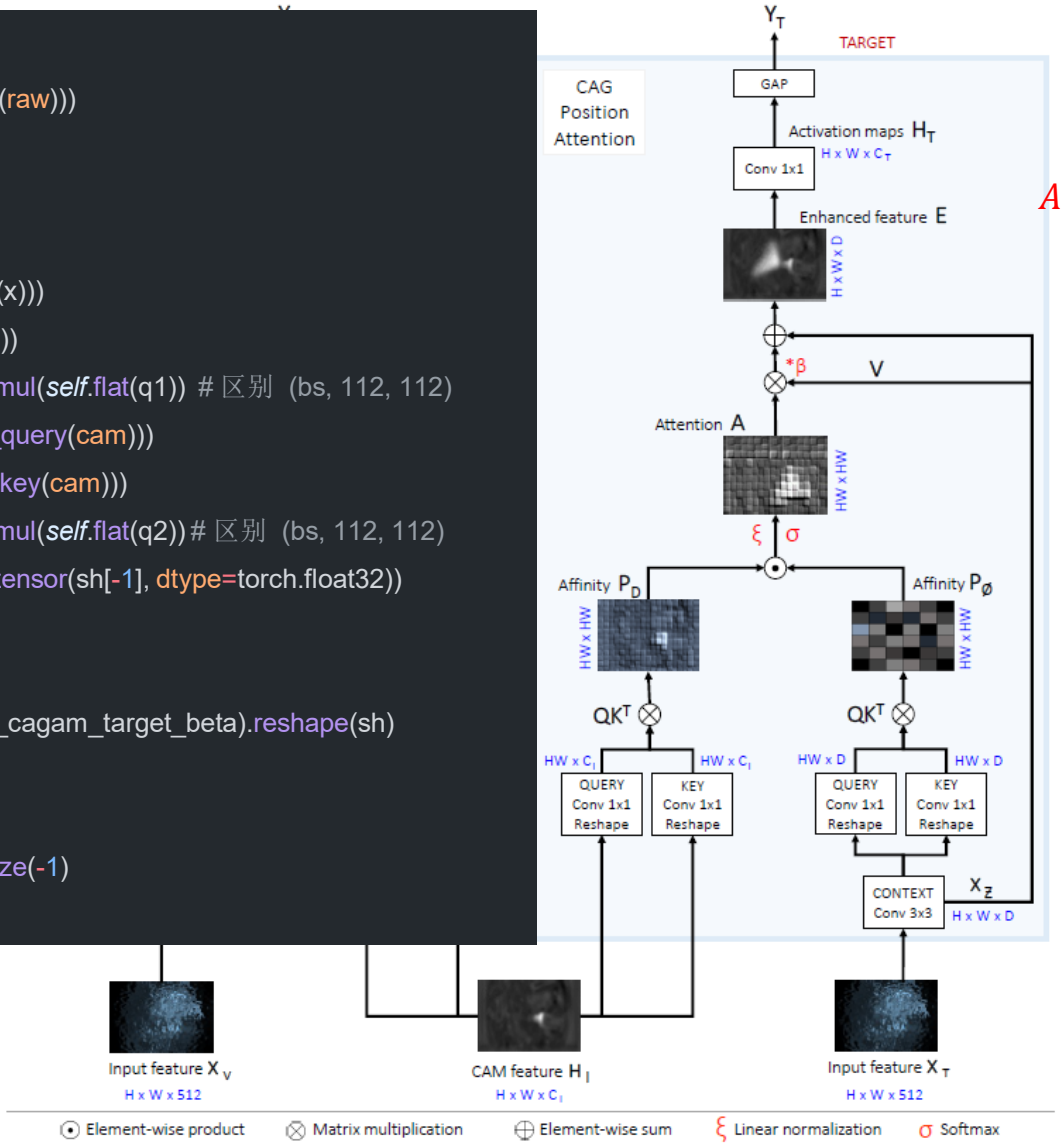

```python
def get_verb(self, raw, cam):
    x = self.elu(self.bn1(self.verb_context(raw)))
    z = x.clone()  # (bs, 6, 8, 14)  clone()操作后的tensor requires_grad=True
    sh = list(z.shape)
    sh[0] = -1
    q1 = self.elu(self.bn2(self.verb_query(x)))
    k1 = self.elu(self.bn3(self.verb_key(x)))
    w1 = self.flat(k1).matmul(self.flat(q1).transpose(-1, -2))
    q2 = self.elu(self.bn4(self.verb_tool_query(cam)))
    k2 = self.elu(self.bn5(self.verb_tool_key(cam)))
    w2 = self.flat(k2).matmul(self.flat(q2).transpose(-1, -2))
    attention = (w1 * w2) / torch.sqrt(torch.tensor(sh[-1], dtype=torch.float32))
    attention = self.soft(attention)  # (bs, 6, 6)
    v = self.flat(z)  # (bs, 6, 112)
    e = (attention.matmul(v) * self.encoder_cagam_verb_beta).reshape(sh)  # (bs, 6, 8, 14)
    e = self.bn6(e + z)
    cmap = self.verb_cmap(e)
    y = self.gmp(cmap).squeeze(-1).squeeze(-1)
    return cmap, y
```

$$A = softmax\left(\frac{P_D P_\phi}{\xi}\right)$$

$$E = \beta(VA) + X_Z$$

研究方法

```
def get_target(self, raw, cam):
    x = self.elu(self.bn7(self.target_context(raw)))
    z = x.clone()
    sh = list(z.shape)
    sh[0] = -1
    q1 = self.elu(self.bn8(self.target_query(x)))
    k1 = self.elu(self.bn9(self.target_key(x)))
    w1 = self.flat(k1).transpose(-1, -2).matmul(self.flat(q1)) # 区别  (bs, 112, 112)
    q2 = self.elu(self.bn10(self.target_tool_query(cam)))
    k2 = self.elu(self.bn11(self.target_tool_key(cam)))
    w2 = self.flat(k2).transpose(-1, -2).matmul(self.flat(q2))# 区别  (bs, 112, 112)
    attention = (w1 * w2) / torch.sqrt(torch.tensor(sh[-1], dtype=torch.float32))
    attention = self.soft(attention)
    v = self.flat(z)
    e = (v.matmul(attention) * self.encoder_cagam_target_beta).reshape(sh)
    e = self.bn12(e + z)
    cmap = self.target_cmap(e)
    y = self.gmp(cmap).squeeze(-1).squeeze(-1)
    return cmap, y
```

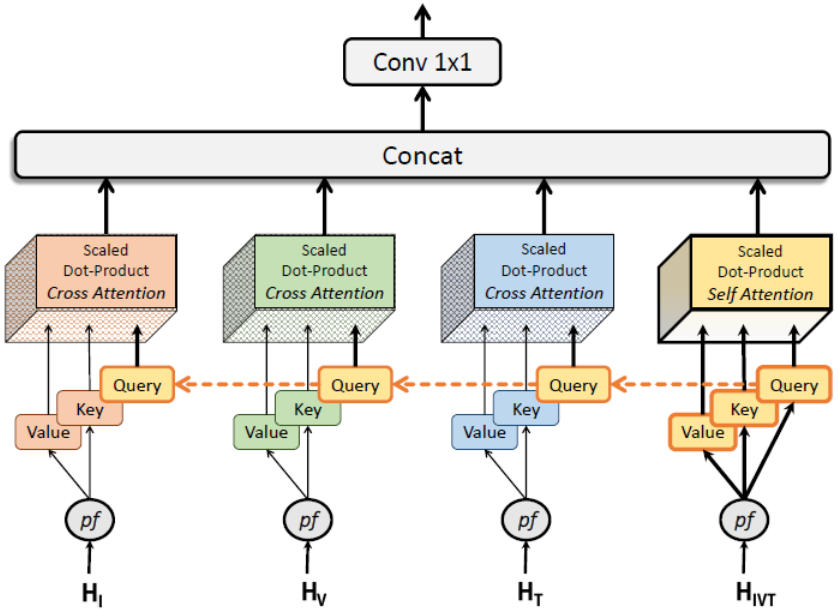


$$A = softmax\left(\frac{P_D P_\phi}{\xi}\right)$$

$$E = \beta(VA) + X_Z$$

研究方法

Decoder-MHMA模块



当每个特征表示为只关注图像场景中的一个组件时，理解它们的底层关系就需要跨组件的交叉注意力。

H_I H_V H_T表示三个组件的语义特征

H_{IVT}表示全局特征

① 先通过映射函数为H_I H_V H_T生成key和value，为H_{IVT}生成query、key、value

$$pf(H)=\begin{cases} Q: & FC\left(DROPOUT\left(GAP\left(\mathbf{H}\right)\right)\right), \\ K: & FC\left(GAP\left(\mathbf{H}\right)\right), \\ V: & CONV\left(\mathbf{H}\right). \end{cases}$$

（为query加上dropout是对于layer=8层来说，避免重复相同的query）

② 结合了自注意力和交叉注意力，通过H_{IVT}的query去匹配H_I H_V H_T的key，得到注意力权重矩阵，再与value相匹配。（全局特征包括三个组件之间的关系，用这个关系去与器械特征匹配）

③ 自注意力是理解自身特征(H_{IVT})表示中的潜在含义和模式；交叉注意力是通过学习每个组件(H_I)的特征是怎么影响三元组特征(H_{IVT})的

研究方法

Decoder

```
class MHMA(nn.Module):
    def __init__(self, depth, num_class=100, num_heads=4, use_ln=False):
        super(MHMA, self).__init__()
        self.concat = nn.Conv2d(in_channels=depth * num_heads, out_channels=num_class, kernel_size=3, padding=1)
        self.bn = nn.BatchNorm2d(num_class)
        self.ln = nn.LayerNorm([num_class, OUT_HEIGHT, OUT_WIDTH]) if use_ln else nn.BatchNorm2d(num_class)
        self.elu = nn.ELU()
        self.soft = nn.Softmax(dim=1)
        self.heads = num_heads

    def scale_dot_product(self, key, value, query):
        dk = torch.sqrt(torch.tensor(list(key.shape)[-2], dtype=torch.float32))
        affinity = key.matmul(query.transpose(-1, -2))
        attn_w = affinity / dk
        attn_w = self.soft(attn_w)
        attention = attn_w.matmul(value)
        return attention

    def forward(self, inputs):
        (X, (k1, v1), (k2, v2), (k3, v3), (q, k, v)) = inputs
        query = torch.stack([q] * self.heads, dim=1)  # [B,Head,D]
        query = query.unsqueeze(dim=-1)  # [B,Head,D,1]
        key = torch.stack([k, k1, k2, k3], dim=1)  # [B,Head,D]
        key = key.unsqueeze(dim=-1)  # [B,Head,D,1]
        value = torch.stack([v, v1, v2, v3], dim=1)  # [B,Head,D,H,W]
        dims = list(value.shape)  # [B,Head,D,H,W]
        value = value.reshape([-1, dims[1], dims[2], dims[3] * dims[4]])  # [B,Head,D,HW]
        attn = self.scale_dot_product(key, value, query)  # [B,Head,D,HW]
        attn = attn.reshape([-1, dims[1] * dims[2], dims[3], dims[4]])  # [B,DHead,H,W]
        mha = self.elu(self.bn(self.concat(attn)))
        mha = self.ln(mha + X.clone())
        return mha
```

研究方法

实验结果

5.4 Quantitative Results on CholecT45 using Cross-Validation Split

Similarly, the benchmarking results on the CholecT45 cross-validation split, presented in Table 8, justifies the use of attention mechanisms for surgical action triplet recognition. The analysis shows that the results obtained on the CholecT45 CV approximates the ones of the CholecT50 CV in all the sub-tasks, justifying its use/sufficiency in the absence of the complete CholecT50 dataset.

Table 8: Benchmark triplet recognition AP (%) on CholecT45 dataset using the official cross-validation split.

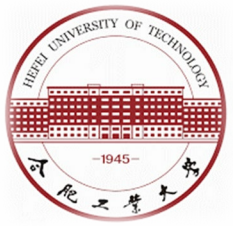
Method (in PyTorch)	Component detection			Triplet association		
	AP_I	AP_V	AP_T	AP_{IV}	AP_{IT}	AP_{IVT}
Tripnet [9]	**89.9±1.0**	59.9±0.9	37.4±1.5	31.8±4.1	27.1±2.8	24.4±4.7
Attention Tripnet [1]	89.1±2.1	61.2±0.6	**40.3±1.2**	33.0±2.9	29.4±1.2	27.2±2.7
Rendezvous [1]	89.3±2.1	**62.0±1.3**	40.0±1.4	**34.0±3.3**	**30.8±2.1**	**29.4±2.8**

据实验过程：组件检测和三元组检测好像并不是同时达到最优，AP_I AP_V AP_T应该是在不同的epoch下达到的最优结果

AP_i: 0.89767
AP_v: 0.62270
AP_t: 0.40138
acc_p: 0.83572

[1]Nwoye C I, Padoy N. Data Splits and Metrics for Method Benchmarking on Surgical Action Triplet Datasets[J]. arXiv, 2022.

Q&A



SW