Temporal Proposal Module for Human Identification at a Distance

Tao Ding[1], Yuchao Yang[1], Shuiwang Li[1], Qijun Zhao[1]

[1]College of Computer Science, Sichuan University

Speaker: Yang Yuchao

Result

User	Accuracy
BeibeiLin	63.0
brl(ours)	54.1
panfengzhang	53.4
ctsu-ca	51.5

Dataset

The training set contains 500 subjects with 10 gait silhouettes sequences for each one.
The test set contains 514 subjects which are different from the training set.

The dataset contains views and walking conditions, such as walking with bag and wearing coat and so on.

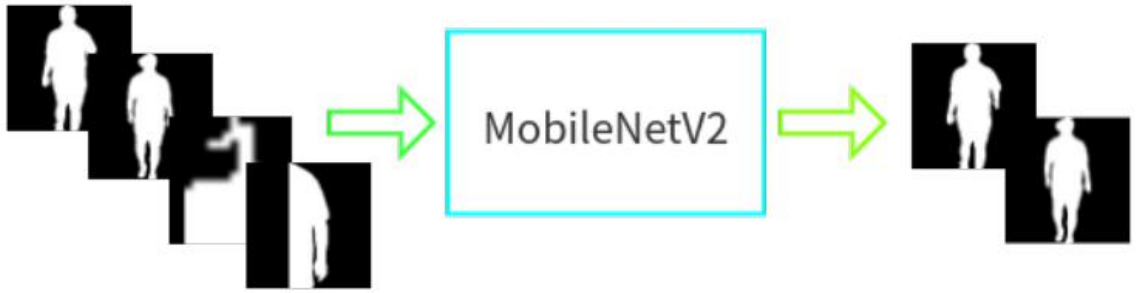
There are quite a lot images of low quality, including images without subject, images with multi-subjects and unclear silhouette.

Motivation

In noisy frames, no gait information can be extracted.

We used a proposal module to select frames that provide gait information and to exclude noisy frames.

Preprocessing



To deal with this, we trained the MobileNetV2 as a classifier to select high quality images from an input sequence in both training and test phases.

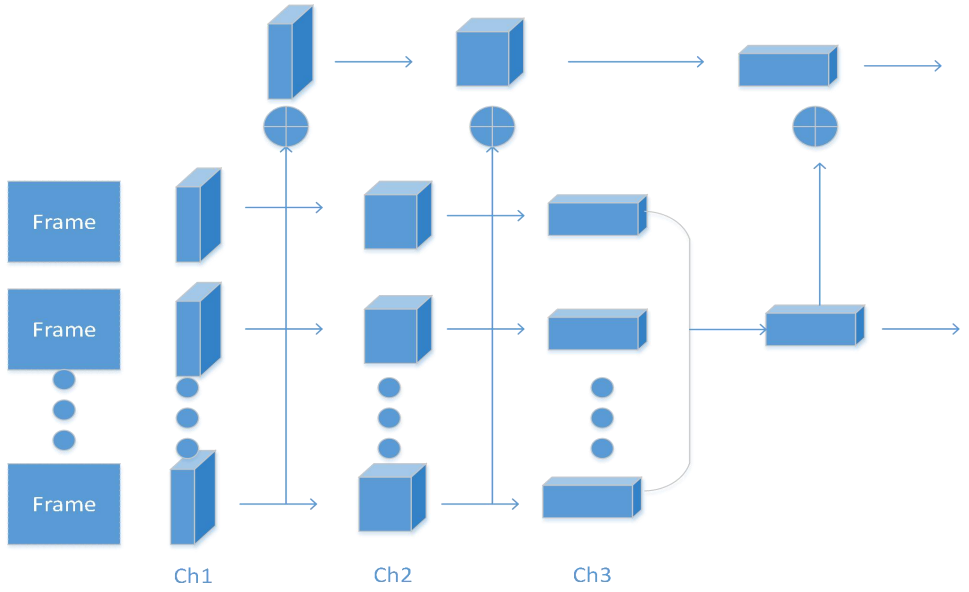
GaitSet

we choose gaitset as our base model.

- Gaitset takes a set of gait silhouette as input.
- After preprocessing, the sequence may no longer be continuous. And Gaitset will not be affected by it.
- Gaitset use set-level feature, not frame-level feature. It did well on CASIA-B.

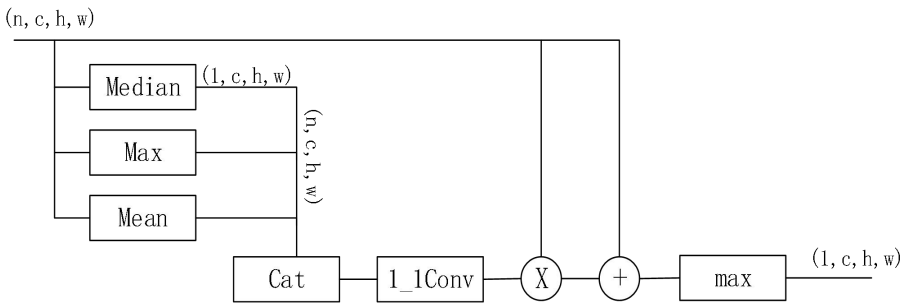
GaitSet

Compared with casia-b, the competition dataset has more subjects. We believe that increasing the number of channels can enhance the feature extraction ability of the backbone. We set Ch1, Ch2, Ch3=64, 128, 256.



Other tries

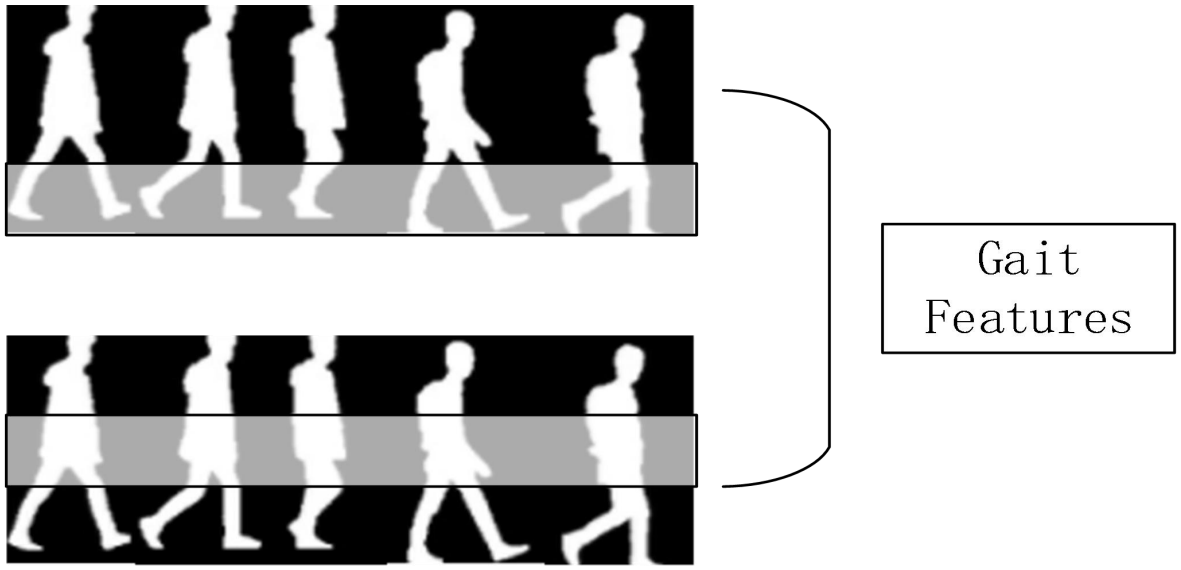
Attention on SP module is mentioned in GaitSet




```python
class SP(nn.Module):
    def __init__(self, in_ch):
        super(SP, self).__init__()
        self.conv1x1 = nn.Conv2d(3 * in_ch, in_ch, 1, bias=False)
    def forward(self, x):

        #if not self.training:
        #    return torch.max(x, dim=1)[0]

        n, s, c, h, w = x.size()

        x_median = torch.median(x, dim=1, keepdim=True)[0]
        x_mean = torch.mean(x, dim=1, keepdim=True)
        x_max = torch.max(x, dim=1, keepdim=True)[0]

        x_median = x_median.repeat(1, s, 1, 1, 1)
        x_mean = x_mean.repeat(1, s, 1, 1, 1)
        x_max = x_max.repeat(1, s, 1, 1, 1)

        y = torch.cat((x_median, x_mean, x_max), dim=2)

        _n, _s, _c, _h, _w = y.size()

        y = y.view(-1, _c, _h, _w)
        y = self.conv1x1(y)
        y = y.view(n, s, c, h, w)
        x = x * y + x
        x = x.view(n, s, c, h, w)
        return torch.max(x, dim=1)[0]
```

Other tries

GaitPart: Temporal Part-Based Model for Gait Recognition



Experiment

Method	Accuracy
GaitSet{[32, 64, 128]}	41.0%
GaitSet{[64, 128, 256]}	50.0%
GaitSet{[64, 128, 256]}+ SP attention	50.5%
GaitSet{[64, 128, 256]}+ MobileNetV2 Selection	54.1%

thanks